SEISAN
EARTHQUAKE ANALYSIS SOFTWARE
FOR WINDOWS, SOLARIS, LINUX and MACOSX

Version 11.0

Editors
Lars Ottemöller\(^{(1)}\) lars.ottemoller@geo.uib.no
Peter Voss\(^{(2)}\) pv@geus.dk
Jens Havskov\(^{(1)}\) jens.havskov@geo.uib.no

(1) Department of Earth Science
University of Bergen
Allgaten 41
5007 Bergen
Norway

(2) Geological Survey of Denmark and Greenland
Oster Voldgade 10
1350 Copenhagen K
Denmark

http://seisan.info

Last corrections September 24, 2020
Cover

The figure on the cover show the output from the new spectrogram command s in MULPLT, see chapter S.10

Citation

If you need to cite this manual: Ottemöller, Voss and Havskov, SEISAN EARTHQUAKE ANALYSIS SOFTWARE FOR WINDOWS, SOLARIS, LINUX and MACOSX, 2020.


Copyright ©2020 Ottemöller, Voss and Havskov.
# Contents

## 1 Introduction

1.1 Changes in the SEISAN alpha version ........................................ 2
1.2 Changes in the new SEISAN version 11 (2018-06-07) .......................... 4
1.3 Changes in SEISAN version 10.5 (2016-11-18) ............................... 5
1.4 SEISAN version 10.4.1 (2016-06-08) .......................................... 5
1.5 SEISAN version 10.4 (2016-03-31) ........................................... 6
1.6 Information about SEISAN online ............................................. 7

## 2 Structure of SEISAN

2.1 Directories ................................................................. 9
2.2 The database ............................................................. 9
   2.2.1 Phase data and hypocenters .................................... 11
   2.2.2 Waveform data and formats ................................... 12
   2.2.3 Continuous waveform data ..................................... 14
2.3 File types used with SEISAN .............................................. 17
2.4 Upper and lower case ..................................................... 18
2.5 Moving data between Sun, Linux, MacOSX and Windows .................. 18

## 3 Installation

3.1 Linux ............................................................................. 20
3.2 MacOSX ................................................................. 23
3.3 Cygwin ................................................................. 23
3.4 Windows ................................................................. 24
3.5 Python ................................................................. 25
3.6 Google Earth .............................................................. 25
3.7 Database directories for your own system, MAKEREA ....................... 26
CONTENTS

5.7 General Work with SEISAN .................................................. 104
5.8 Graphics in SEISAN ............................................................. 105
5.9 Logging in SEISAN .............................................................. 105
5.10 Known problems in SEISAN ................................................. 106

6 Description of Programs and Commands .................................. 109

7 Hypocenter location programs: HYPOCENTER, HYPO71 and HYPOINVERSE 111
7.1 The hypocenter program, HYP ............................................. 111
7.1.1 Phases ............................................................................. 111
7.1.2 Azimuth and single station location ................................... 111
7.1.3 Magnitudes ...................................................................... 112
7.1.4 Use of S-P and L-S differences ........................................ 117
7.1.5 Global event location ....................................................... 117
7.1.6 Criteria for a solution and weighting ................................ 118
7.1.7 Eliminating outliers ......................................................... 119
7.1.8 Determining which travel time software is used ................. 120
7.1.9 Starting location ............................................................. 120
7.1.10 Preventing a local minimum ........................................... 121
7.1.11 Alternative model .......................................................... 121
7.1.12 Location by grid search .................................................. 121
7.1.13 Using HYP to determine crustal structure ....................... 122
7.1.14 Running HYP ............................................................... 123
7.1.15 Station and model files .................................................. 125
7.1.16 RESET TEST parameters ............................................ 128
7.1.17 HYP output ................................................................. 131
7.1.18 Problems ....................................................................... 133
7.2 HYPO71 (Sun only) ............................................................. 133
7.3 The Hypoinverse program, HYPINV ...................................... 137
7.4 Plugin for the iLoc location algorithm, ILOC_SEISAN (Linux only) 141

8 Trace plotting, phase picking and spectral analysis, MULPLT .......... 143
8.1 MULPLT main functions ....................................................... 144
8.2 Use of MULPLT from EEV and SE ...................................... 146
8.3 Working with many channels in MULPLT ............................... 147
# CONTENTS

8.4 Continuous plotting of one channel .............................. 149
8.5 Commands in MULPLT, overview .................................. 150
8.6 Registering new events into SEISAN ............................... 162
8.7 Phase picking, amplitude, weight and polarity ...................... 164
8.8 Data manipulation commands ...................................... 170
8.9 Spectral analysis, s(Spec) ......................................... 175
8.10 Spectrogram ....................................................... 182
8.11 Particle motion plots .............................................. 187
8.12 Setting MULPLT parameters, MULPLT.DEF ....................... 189

9 Distance trace plot with GMT, TRACE_PLOT (Unix only) .......... 197

10 Plotting epicenters .................................................. 201
   10.1 EPIMAP .......................................................... 201
   10.2 MAP, running EPIMAP without questions ......................... 209
   10.3 W_EMAP, Version Windows based map program .................. 209
   10.4 MAPG, a simple automatic GMT map ............................ 211
   10.5 GMAP, Plotting epicentres with Google maps or Google Earth .. 215
      10.5.1 The simple GMAP ........................................... 215
      10.5.2 The advanced GMAP ...................................... 216
      10.5.3 The automatic GMAP ..................................... 222
   10.6 SEIS2VIEWER, Plotting hypocentres in 3D ...................... 224

11 Searching in the database, SELECT and others .................... 227
   11.1 SELECT ........................................................ 227
       11.1.1 Input parameters for SELECT ............................. 230
       11.1.2 Option for codaq ........................................ 232
       11.1.3 Historical data .......................................... 233
       11.1.4 Select with input from the prompt line .................. 234
   11.2 Searching for text string in nordic files, SELECTC ............ 234
   11.3 Getstressdrop ................................................ 235

12 Extracting events from the database, COLLECT .................... 239

13 Inserting events into the database, SPLIT ......................... 241
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 Updating final locations in database, UPDATE and UPD</td>
<td>243</td>
</tr>
<tr>
<td>15 Using filenr.lis, DIRF and DELF</td>
<td>245</td>
</tr>
<tr>
<td>16 Making a bulletin, BUL</td>
<td>247</td>
</tr>
<tr>
<td>17 Reports and statistics</td>
<td>249</td>
</tr>
<tr>
<td>17.1 REPORT, extract data from CAT file</td>
<td>249</td>
</tr>
<tr>
<td>17.2 NORHEAD, making a compact Nordic file from a Nordic file</td>
<td>250</td>
</tr>
<tr>
<td>17.3 STATIS, statistics of databases</td>
<td>251</td>
</tr>
<tr>
<td>17.4 CATSTAT, statistics of catalogs</td>
<td>252</td>
</tr>
<tr>
<td>17.5 SWARM, finding earthquake swarms</td>
<td>253</td>
</tr>
<tr>
<td>17.6 STATSTAT, number of events per seismic station in catalog</td>
<td>253</td>
</tr>
<tr>
<td>17.7 LSQ, plotting a linear relation or a curve</td>
<td>253</td>
</tr>
<tr>
<td>18 Waveform file management tools</td>
<td>255</td>
</tr>
<tr>
<td>18.1 APPEND, Append two or more waveform files</td>
<td>256</td>
</tr>
<tr>
<td>18.2 GET_ARC, extract waveform files from the archive corresponding to S-files and register the name in the S-file</td>
<td>263</td>
</tr>
<tr>
<td>18.3 GET_WAV, get listing of available waveform and response files</td>
<td>265</td>
</tr>
<tr>
<td>18.4 RDSEED_MANY, chop up seed file</td>
<td>266</td>
</tr>
<tr>
<td>18.5 RESAMP, resampling waveform files</td>
<td>266</td>
</tr>
<tr>
<td>18.6 SEIASC, converting SEISAN waveform files to or from ASCII</td>
<td>267</td>
</tr>
<tr>
<td>18.7 SEICUT, extract part of a waveform file</td>
<td>267</td>
</tr>
<tr>
<td>18.8 SEIDEL, splitting a SEISAN binary file into 2 files</td>
<td>267</td>
</tr>
<tr>
<td>18.9 SEISEL, splitting and merging SEISAN readable binary files</td>
<td>268</td>
</tr>
<tr>
<td>18.10 SEELSEI, searching headers in SEISAN waveform files</td>
<td>268</td>
</tr>
<tr>
<td>18.11 P_ALIGN: Time shifting waveform data to align on P or S-phase arrival times</td>
<td>269</td>
</tr>
<tr>
<td>18.12 WAVFIX, fixing time correction and channel names in waveform file headers and make standard file names</td>
<td>269</td>
</tr>
<tr>
<td>18.13 WAVFULLNAME</td>
<td>272</td>
</tr>
<tr>
<td>19 File conversion and modification programs</td>
<td>273</td>
</tr>
<tr>
<td>19.1 Nordic to QuakeML converter</td>
<td>299</td>
</tr>
<tr>
<td>20 Signal processing programs</td>
<td>303</td>
</tr>
<tr>
<td>20.1 SAC2000</td>
<td>303</td>
</tr>
</tbody>
</table>
# CONTENTS

## 21 Automatic routines in SEISAN

21.1 AUTOPIC and AUTO ............................................. 305
21.2 AUTOPHASE, automatic phase picking .......................... 310
21.3 AUTOSIG .......................................................... 315
21.4 AUTOMAG ......................................................... 319
21.5 PLOTSPEC .......................................................... 325
21.6 Detection program for continuous data, CONDET .............. 328

## 22 Calculating b-value, BVALUE

## 23 Fault plane solution

23.1 FOCMEC ........................................................... 335
  23.1.1 Use of amplitudes ............................................ 336
  23.1.2 Automatic reading of amplitudes ............................ 337
  23.1.3 Polarity selection ............................................. 337
  23.1.4 Local earthquakes ............................................ 338
  23.1.5 Program operation ............................................ 338
  23.1.6 Making synthetic amplitudes and polarities to test FOCMEC and HASH .......................... 341
23.2 FPFTT ............................................................. 342
23.3 HASH ............................................................... 344
  23.3.1 Running HASH from EEE ...................................... 345
  23.3.2 Explanations of output ...................................... 346
23.4 PINV .............................................................. 347
23.5 Fault plane solution program to use and some advice .......... 348
23.6 Composite fault plane solution using any of the programs FOCMEC, FPFTT, HASH and PINV ...................... 349
23.7 Moment tensor inversion program, INVRAD ...................... 350
23.8 Plotting fault plane solutions ................................... 352
23.9 FOC: Plot many FPS, stress inversion and Rose diagram .... 352
23.10 PLOTFOC, plotting fault plane solution without need for station information ............................. 357
23.11 MOPAD, plotting a moment tensor solution and more ........ 357

## 24 Plotting and changing of polarities, PLOTPOLARITY ......... 361

## 25 Measuring amplitude ratios, AUTORATIO

25.1 Plotting amplitude ratios, PLOTRATIO .......................... 366
<table>
<thead>
<tr>
<th>CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>26 Waveform inversion</strong></td>
</tr>
<tr>
<td>26.1 Moment Tensor inversion in SEISAN</td>
</tr>
<tr>
<td><strong>27 Calculation of coda q, CODAQ, and related programs</strong></td>
</tr>
<tr>
<td>27.0.1 Input</td>
</tr>
<tr>
<td>27.0.2 Program operation</td>
</tr>
<tr>
<td>27.0.3 Output files</td>
</tr>
<tr>
<td>27.0.4 General recommendations and example of test data</td>
</tr>
<tr>
<td>27.1 Program CODAQ_AREA</td>
</tr>
<tr>
<td>27.2 Program CODAQ_POLYGON</td>
</tr>
<tr>
<td>27.3 Program QSTAT</td>
</tr>
<tr>
<td>27.4 Program AVQ, average Q-relations</td>
</tr>
<tr>
<td><strong>28 Merge events in SEISAN and compare catalogs</strong></td>
</tr>
<tr>
<td>28.1 Merge events near in time ASSOCI</td>
</tr>
<tr>
<td>28.2 Merge events near in time and delete duplicate phases, MERGE_S</td>
</tr>
<tr>
<td>28.3 Merge events near in time, distance, depth and magnitude ASSO</td>
</tr>
<tr>
<td>28.4 Catalogue merging and comparison with merge_seisan.pl</td>
</tr>
<tr>
<td>28.5 COMPARE_HYP, compare hypocenters of two cat-files</td>
</tr>
<tr>
<td><strong>29 Making synthetic seismograms</strong></td>
</tr>
<tr>
<td><strong>30 Calculation and plotting of travel times</strong></td>
</tr>
<tr>
<td>30.1 IASPEI travel time software, program TTIM</td>
</tr>
<tr>
<td>30.2 Calculation of travel times for layer and gradient model, TTLAYER</td>
</tr>
<tr>
<td>30.3 Calculating of local phases: TTLOCAL</td>
</tr>
<tr>
<td>30.4 Plotting of travel times, TTPlot</td>
</tr>
<tr>
<td>30.5 IASP, travel times for MULPLT</td>
</tr>
<tr>
<td><strong>31 Inversion for Q_Lg, QLG</strong></td>
</tr>
<tr>
<td><strong>32 Wadati</strong></td>
</tr>
<tr>
<td><strong>33 Calculating spectra, the SPEC program</strong></td>
</tr>
<tr>
<td><strong>34 Seismic risk related programs</strong></td>
</tr>
<tr>
<td>34.1 useful programs for hazard</td>
</tr>
<tr>
<td>Section</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>34.2 CRISIS2012</td>
</tr>
<tr>
<td>34.3 SEISAN EXPLORER</td>
</tr>
<tr>
<td>34.4 CLUSTER</td>
</tr>
<tr>
<td><strong>35 Magnitude relations, MAG</strong></td>
</tr>
<tr>
<td><strong>36 ML inversion, MAG2</strong></td>
</tr>
<tr>
<td>36.1 MAGSTAT</td>
</tr>
<tr>
<td><strong>37 Explosion filtering, EXFILTER</strong></td>
</tr>
<tr>
<td><strong>38 Inversion of travel time data and joint hypocenter determination</strong></td>
</tr>
<tr>
<td>38.1 VELEST</td>
</tr>
<tr>
<td>38.2 NOR2DD</td>
</tr>
<tr>
<td>38.3 hypoDD</td>
</tr>
<tr>
<td>38.3.1 hypoDD example</td>
</tr>
<tr>
<td>38.4 NOR2SIMULPS, NOR2SIMULR</td>
</tr>
<tr>
<td>38.5 NOR2JHD, PUJOL</td>
</tr>
<tr>
<td><strong>39 Analysis of volcanic earthquakes</strong></td>
</tr>
<tr>
<td><strong>40 FK Analysis</strong></td>
</tr>
<tr>
<td><strong>41 Instrument response</strong></td>
</tr>
<tr>
<td>41.1 Create instrument response files, RESP</td>
</tr>
<tr>
<td>41.2 Examples of main response files from seismometers and accelerometer</td>
</tr>
<tr>
<td>41.3 SEED response</td>
</tr>
<tr>
<td>41.4 SEED response to GSE, SEEDRESP2GSE</td>
</tr>
<tr>
<td>41.5 GSE response to SEED, GSERESP2SEED</td>
</tr>
<tr>
<td><strong>42 Macroseismic data in SEISAN</strong></td>
</tr>
<tr>
<td><strong>43 Correlation of waveform signals, CORR and detection of event clusters XCLUST</strong></td>
</tr>
<tr>
<td><strong>44 Programming in SEISAN and list of files in SEISAN distribution</strong></td>
</tr>
<tr>
<td>44.1 Programmers guide and some test programs</td>
</tr>
<tr>
<td>44.2 Routines for Geodetic computations (WGS84)</td>
</tr>
<tr>
<td>44.3 CONTENTS OF PRO, LIB, INC, INF, COM, DAT, SUP, ISO and PIC DIRECTORIES</td>
</tr>
</tbody>
</table>
A The Nordic format 537
B The SEISAN waveform file format 551
C Response file formats used in SEISAN 559
C.1 SEISAN response format 559
C.2 GSE response format 561
C.3 SAC format 563
C.4 RESP format 563
D Some general subroutines in SEISAN in LIB 565
E Changelog 573
E.1 SEISAN version 10.3 (2015-01-17) 573
E.2 SEISAN version 10.2 (2014-09-20) 573
E.3 SEISAN version 10.1 (2014-06-19) 574
E.4 SEISAN version 10.0 (2013-06-11) 574
E.5 SEISAN version 9.1 (2013) 575
E.6 SEISAN version 9.0.1 (2011) 576
E.7 SEISAN version 9.0 (2011) 576
E.8 SEISAN version 8.3 (2010-06-05) 577
E.9 SEISAN version 8.2.1 (2008-10-10) 578
E.10 SEISAN version 8.2 (2008) 578
Chapter 1

Introduction

The SEISAN seismic analysis system is a complete set of programs and a simple database for analyzing earthquakes from analog and digital data. With SEISAN it is possible using local and global earthquakes to enter phase readings manually or pick them with a cursor, locate events, edit events, determine spectral parameters, seismic moment, azimuth of arrival from 3-component stations and plot epicenters. The system consists of a set of programs tied to the same database. Using the search programs it is possible to use different criteria to search the database for particular events and work with this subset without extracting the events. Most of the programs can operate both in a conventional way (using a single file with many events), or in a database manner. Additionally, SEISAN contains some integrated research type programs like coda Q, synthetic modeling, moment tensor inversion and a complete system for seismic hazard calculation.

The data is organized in a database like structure using the file system. The smallest basic unit is a file (the S-file) containing original phase readings (arrival times, amplitude, period, azimuth, and apparent velocity) for one event. The name of that file is also the event ID, which is the key to all information about the event in the database. Although the database in reality only consists of a large number of sub-directories and files (all of which the user has access to), the intention is that by using the surrounding software, the user should rarely need to access the files directly, but rather do all work from the user’s own directory. Included with SEISAN there is a test data set, a training document using the test data of many events and a tutorial using just the 2 events included with the software. In addition, there is a document illustrating seismogram analysis using SEISAN. See chapter 5. To get an idea of how SEISAN works, new users can take a look at a SEISAN webinar that was arranged by IRIS in 2014 (see https://www.youtube.com/watch?v=KJH3ktGtL_K0 or file seisan_webinar_iris_2014.mov on the SEISAN ftp server).

SEISAN runs under Sun Solaris, Linux, MacOSX and Windows. The programs are mostly written in Fortran, a few in C and almost all source codes is given, so the user should be able to fix bugs and make modifications. The programs have been compiled and linked with system compilers and linkers on SUN, GNU compiler on Linux, Windows and MaxOSX. For graphics, X is used on Unix systems and DISLIN (www.dislin.de) used under Windows. No format conversion is needed to move data files (binary and ASCII) between the systems if one of the standard formats (SEISAN, GSE2.0, SEED, SAC, GURALP) is used. The GUI Seisan Explorer is written in Qt and is used on Linux, MacOSX and Windows.

BE AWARE: A SEISAN S-file must not contain symbols requiring 2 bytes or more (the case with some country letters). The system will crash. A fix will be made in next version.
CHAPTER 1. INTRODUCTION

This manual resides in the directory INF (see below), when the system has been implemented on your computer. The file is called seisan.pdf (Adobe PDF).

The SEISAN system is built of programs made by many different individuals without whom it would never have been possible to make SEISAN. Acknowledgement is made throughout this manual where appropriate or in the acknowledgement section at the end. SEISAN now contains so many programs that when a new version is released, it is not possible to check all the options in all programs and we rely on the user to help finding the bugs, please report!

SEISAN is freely available for all non-commercial use.

In this manual names of computer programs are given with capital letters, names of files and command line options are given by typewriter font.

1.1 Changes in the SEISAN alpha version

- **NOR2QML** - A online service to convert NORDIC format to and from QuakeML, see page 282. There is also a Java script included which does the same called nor2qml or qml2nor.
- **SELECT** has changed waveform name search to a general search of a text string in whole file, optionally case independent. Program SELECTC has now been removed since the new SELECT can do the same.
- **HYP** can optionally do a grid search (3D) at end of iterations to find if any local minimum. If so it then starts again iterating from the local minimum. Controlled by RESET TEST parameters.
- A plugin for the iLoc program has been implemented, see page 141.
- A new program CHANGE_RESP_NAMES to fix RESP file names for missing or wrong location and network codes.
- Automatic amplitude can now also be used in multi trace mode.
- **SELECT** can use moment tensor as selection criteria.
- HypoDD has been included, all platforms. A tutorial is included.
- Program WAVASC has been added, makes 2 column ASCII file from any waveform file format.
- Program MERGE_S has been added to merge S-file catalogs and remove duplicate phases.
- Program IMSNOR has been added. Program for converting IMS and ISF format files, more complete than NORIMS. It also supports the new ISF2 format.
- EEV option US added to make and update ID lines
- Program VELNOR has been added, for converting Velest cnv file to Nordic.
- Program NOR2NEW, converts between new and old Nordic formats.

New Nordic Format

The Nordic format has not included network and location codes. More and more station now use these codes and now two different stations can have the same station code and then they have to be differentiated by network code and possibly location code. A significant weakness with the format is that component
1.1. CHANGES IN THE SEISAN ALPHA VERSION

codes are 2 letters which should have been 3 so in the new format they are 3. Another problem when
data is merged from different sources is to know where the data come from. We have therefore, for each
phase, included the possibility to include both agency and author of the phase.

The philosophy of the new format is that each type of observation used in S-files (phase, amplitude, coda,
back azimuth) must occupy one line with a corresponding phase. This has been an optional case also
before for amplitudes and back azimuth, now it is a must. The amplitude phases have the same name as
before like IAML, the back azimuth (and apparent velocity) are BAZ, possibly with a phase name like
BAZ-P. The coda is END (IASPEI standard). Below is an example, the Appendix has a full description.

<table>
<thead>
<tr>
<th>STAT</th>
<th>COM</th>
<th>NTLO</th>
<th>IPHASE</th>
<th>W</th>
<th>HHHM</th>
<th>SS.SSS</th>
<th>PAR1</th>
<th>PAR2</th>
<th>AGA</th>
<th>OPE</th>
<th>AIN</th>
<th>RES</th>
<th>W</th>
<th>DIS</th>
<th>CAZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>BER</td>
<td>SHZ</td>
<td>NS00</td>
<td>IP</td>
<td>32</td>
<td>1325</td>
<td>48.120</td>
<td>C</td>
<td>1325</td>
<td>BER</td>
<td>1.0</td>
<td>-1.9801061.00</td>
<td>0.00</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASK</td>
<td>SHZ</td>
<td>NS00</td>
<td>ES</td>
<td>32</td>
<td>1325</td>
<td>48.070</td>
<td>BER</td>
<td>1325</td>
<td>jh</td>
<td>-1.021071.10</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASK</td>
<td>SHZ</td>
<td>NS00</td>
<td>AMP</td>
<td>32</td>
<td>1325</td>
<td>48.070</td>
<td>333.3</td>
<td>2.20</td>
<td>2.50</td>
<td>30</td>
<td>71.10</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASK</td>
<td>SHZ</td>
<td>NS00</td>
<td>IAML</td>
<td>32</td>
<td>1325</td>
<td>50.900</td>
<td>111.0</td>
<td>0.30</td>
<td></td>
<td></td>
<td>71.10</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NRA0</td>
<td>SHZ</td>
<td>BB01</td>
<td>IPn</td>
<td>32</td>
<td>1326</td>
<td>19.090</td>
<td>NA0</td>
<td>1326</td>
<td>hhh</td>
<td>50.0 -.05010368.0</td>
<td>0.00</td>
<td>72</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NRA0</td>
<td>S</td>
<td>Z</td>
<td>END</td>
<td>32</td>
<td>1326</td>
<td>19.090</td>
<td>333.3</td>
<td>0.2</td>
<td>368.0</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NRA0</td>
<td>S</td>
<td>Z</td>
<td>BAZ-P</td>
<td>32</td>
<td>1326</td>
<td>19.090</td>
<td>256.9</td>
<td>6.9</td>
<td>0.1</td>
<td>368.0</td>
<td>72</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUD</td>
<td>BHZ</td>
<td>DK00</td>
<td>IP</td>
<td>32</td>
<td>1326</td>
<td>38.120</td>
<td>C</td>
<td>1326</td>
<td>DNK</td>
<td>-1.9801061.00</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUD</td>
<td>BHE</td>
<td>DK00</td>
<td>ES</td>
<td>32</td>
<td>1326</td>
<td>48.070</td>
<td>DNK</td>
<td>1326</td>
<td>pv</td>
<td>-1.021071.10</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The new columns are NT and LO for network and location respectively. AGA and OPE are agency and
operators. PAR1 and PAR2 are the observation associated with the phase which are:

Seismic phase: polarity
Amplitude: amplitude and period
END: coda length
BAZ: back azimuth and apparent velocity

The residual (RES) is now residual for the observation if any, like degree for BAZ, seconds for phase,
magnitude for amplitude and coda.

The user can change the format to use in SEISAN.DEF, parameter NEW_NORDIC_FORMAT. The
default, for now, in the distribution is that the format is set to the old Nordic format (0) and everything
will work as before. Setting to 1 old and new format can be read but new will be written and setting to
2 only new for a can be used.

It is possible to convert from new to old and vise versa by program NOR2NEW.

The majority of programs have been changed to be able to work with the new format, if not it will be
noted.

There are several programs that work with the SEISAN 4 character component names, particularly
conversion programs. These have not been changed to 3 character components since they will also work
with the 3 character components, leaving the 3. character blank. When converting from 2 character to
3 character component codes, the 2. character is left blank. Also many programs can only work with
station and component and will not, at the moment, be converted to also use network and locations (e.g.
CODAQ).

Old to new components, NT and LO in MULPLT
When old data is transferred to new format there is only component e.g. S Z and no location and network
code. If the original data has this information, MULPLT is still able to plot the phases since:

- if component code 2 is blank, no check for 2. component code.
CHAPTER 1. INTRODUCTION

- check for location and network is disabled if they are both blank.

If new phases are read and NT or LO are present, the component code must then have 3 characters.

Response files. The SEISAN naming of response files do not have NTLO so they cannot distinguish same station names and different NTLO. This will probably also be fixed but for now the problem can be solved by using RESP response files.

So far the headers in the S-file have not been changed but there is plan to also change the SPEC lines to include full station, component and NTLO information.

For most users, there is probably not pressing need to start using the new format, but eventually all should use it. The plan is to continue to distribute a SEISAN which is able to use the old format.

Warning (by 2020-09-24): The manual is still under modification so much information about the new format will be missing.

1.2 Changes in the new SEISAN version 11 (2018-06-07)

- SEISAN EXPLORER has seen a major upgrade with the location interface and the integration with MULPLT. It is now possible to work with picking phases in MULPLT and immediately see the results in the location window. In addition there are other major changes.
- Bug in MULPLT Continuous on screen has been fixed, the bug was introduced in version 10.5
- MSCUT: Can now be used to cut large miniseed files into files with a shorter duration in minutes, where 60 modulo duration is equal to zero.
- EEV has new options: ste and std for setting and unsetting start flags, ep to see the print.out file
- EEV can plot fault plane solution on a map option MAPF
- Include program MOPAD to plot moment tensor solutions, option fm in EEV
- A new program PLOTFOC to plot fault plane solutions from a CAT-file without need for relocating event. Also in EEV with command 'foo'.
- A new option in EEV to compare the P and T axis of two fault plane solutions, fd
- A new version of the AUTO program that can run anyone or a combination of the programs AUTOPHASE, AUTOPIC, HYP, AUTOMAG, AUTORATIO, FPFIT and HASH in loop of many events
- FOCMEC has been updated with the latest FOCMEC program from Snoke and it can now use relative polarity weight and has a better treatment of amplitudes near nodal planes. The old version can still be used (f_old) in EEV.
- A new sample program, SAMPLE_READ_WRITE_S_C, written in c++, can read and write S-files in C++ and provides a structure with all the variables.
- An updated Seisan_explorer with graphical locate interface.
- Spectrograms can be made from MULPLT or with program SPECTROGRAM, see section 8.10
- A new program MAP which runs EPIMAP without questions, only the input file.
1.3. Changes in SEISAN version 10.5 (2016-11-18)

- A new program MAPG which makes a map in GMT without questions, only the input file or database is needed. Also plots fault plane solutions. Also works with command MAPG in EEV.
- New options in EEV: STE: Set or unset start location, STD: Set or unset start location depth, FIXO: Fix origin time.
- A new program, GET,WEB,LOCATIONs to download hypocenter information from web sites, so far implemented with USGS.
- New magnitude scale MN or mb(Pn/Sn) for Pn and Sn at regional scale is implemented.

1.4 SEISAN version 10.4.1 (2016-06-08)

- Fixing reading of SEED files on Windows
- Updating the sfiles is now possible from MULPLT, option appears after locate command
- A new section has been added to the manual regarding response files
- The order of which a response file is found has been changed. SEISAN will now first look for a SEED RESP file that is valid for the selected data, before it looks for a matching SEISAN/SAC/GSE response file. Previously, SEED RESP files was checked last. The change in order is due to the addition of SEED network and location codes and since RESP files include an expiration date, which SEISAN/SAC/GSE response files does not.
- DATASELECT added to distribution (see SUP folder), DATASELECT is replacing MSROUTER since IRIS is no longer supporting MSROUTER (to use DATASELECT see page 77).
1.5 SEISAN version 10.4 (2016-03-31)

- SEISAN can now also read SEED real numbers, but not write SEED real numbers
- HYP has a new option for finding and eliminating travel time outliers, EEV commands lr and ur
- Logging of MULPLT continuous mode, see page 149
- Logging of EEV commands: (see page 105)
  - When comment is inserted (COM command)
  - When event is registered (REG or PUT command)
  - When event is updated (UPDATE command)
  - When event is duplicated (DUP command)
  - When event is moved or copied to another database (C command)
  - When event type is changed (R command)
  - When sfile has been edited (E command)
  - When event is deleted (D command)
- Iasp command in MULPLT use local model
- Iasp all available waveform channels used
- Definition of virtual networks for archive in SEISAN.DEF
- Call to (external) spectrogram from MULPLT (command 'E')
- GSERESP2SEED added to distribution
- MSROUTER added to distribution
- New option in CODAQ and new coda Q analysis program
- GET_STAT command to print station coordinates
- New Locator in SEISAN EXPLORER
- Update CODAQ
- New program to analyze coda Q results, QSTAT
- AUTOREG now runs with flag to set location model indicator
- Magnitude calculation has changed a bit in terms of distance limits, so a few magnitudes might have changed
- New program GET_ARC to extract waveform files from an archive based on S-files and add the waveform name to the S-file.
- New option arx in EEV to use GET_ARC
- New program ARC_DEL that will remove ARC lines in sfiles
- New option arcdel in EEV to use ARC_DEL
1.6. INFORMATION ABOUT SEISAN ONLINE

- New option in EEV to add ARC lines in sfiles for virtual nets, see ARC .
- New options in EEV to enter one of a set of fixed comments, command ic.
- New command in EEV to fix, unfix of enter fixed depth, fix
- MULPLT command u for rotate is now U.
- MULPLT can now go from one station to the next in three channel mode.
- New command in EEV to plot in single channel mode using all defaults, poo.
- New programs to convert Nordic input to simulps and simulr
- New version of WAVFIX that accept all SEISAN used waveform formats and write out in MiniSeed

Older changes are listed in Appendix E

1.6 Information about SEISAN online

SEISAN homepage
The URL address where SEISAN and related software can be found is: https://www.uib.no/rg/geodyn/artikler/2010/02/software
Here you can find information on the latest changes in SEISAN, access the online manual, download the software and much more.

SEISAN anonymous ftp server
Seisan is available from the following ftp server:
ftp.geo.uib.no (or 129.177.55.28)
Login: ftp
Password: your email address
The files are stored in the directory /pub/seismo/SOFTWARE/SEISAN.

SEISAN mailing lists
There is a mailing list, which is set-up to improve the exchange of information and questions on SEISAN. We strongly recommend that all users subscribe to the SEISAN list. The list is:
seisan@geo.uib.no
The purpose of the list is:
To subscribe to the list, send an email to seisan-join@uib.no or to seisan-request@uib.no with subject/body subscribe.
Subscription can also be done online via http://mailman.uib.no/listinfo/seisan. As a member of the list, it is possible to look through the archive (since 2008) for questions and answers on SEISAN. Anyone is welcome to reply to questions and a response to a question should be send to the complete mailing list.
Chapter 2

Structure of SEISAN

2.1 Directories

The whole SEISAN system is located in subdirectories residing under the main directory SEISMO. For more details, see chapter 3 on installation. The system contains the following main subdirectories:

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REA</td>
<td>Earthquake readings and full epicenter solutions in a database</td>
</tr>
<tr>
<td>WOR</td>
<td>The users work directory, initially empty</td>
</tr>
<tr>
<td>TMP</td>
<td>Temporal storage of files, initially empty</td>
</tr>
<tr>
<td>PRO</td>
<td>Programs, source code and executables</td>
</tr>
<tr>
<td>LIB</td>
<td>Libraries and subroutines</td>
</tr>
<tr>
<td>INC</td>
<td>Include files for programs and subroutines in PRO and LIB</td>
</tr>
<tr>
<td>COM</td>
<td>Command procedures</td>
</tr>
<tr>
<td>DAT</td>
<td>Default and parameter files, e.g. station coordinates</td>
</tr>
<tr>
<td>WAV</td>
<td>Digital waveform data files</td>
</tr>
<tr>
<td>CAL</td>
<td>System calibration files</td>
</tr>
<tr>
<td>INF</td>
<td>Documentation and information</td>
</tr>
<tr>
<td>ISO</td>
<td>Macroseismic information</td>
</tr>
<tr>
<td>SUP</td>
<td>Supplementary files and programs</td>
</tr>
</tbody>
</table>

In the following, the above subdirectories will mostly be called directories to avoid always referring to SEISMO. All directories use capital letters, however this only makes a difference in the Unix versions. The directory structure is used as a tree like structure for quick access to individual files in the REA directory, which therefore will appear as a simple database to the user. The next section is a description of the database directories; the other directories are described in chapter 44. Figure 2.1 shows the tree structure of SEISAN.

2.2 The database

The database of SEISAN consists of the two directories REA and WAV. The REA directory and its subdirectories contain readings and source information while all waveform data is normally in the directory WAV (see 2.2.2) with no subdirectories. Optionally WAV can also be divided into a similar subdirectory structure, see 2.2.2, which is useful when storing continuous data in particular. The DELET database
Figure 2.1: Structure of SEISAN. Note that BERGE under WAV is optional and DELET (not shown) under REA has a similar directory structure as e.g. NAO.
contains all events deleted from any of the databases (here BERGE/BER and NAO). Filenames are identical between all platforms.

### 2.2.1 Phase data and hypocenters

The REA directory contains phase readings and derived source information like hypocenters, fault plane solutions etc. The REA directory has one or several subdirectories corresponding to separate databases (see Figure 2.1 for an example with two databases). The database names can have between 3 and 5 characters. If less than 5 characters are used, the character ‘.’ is added in the file system to make it 5. The user does not have to put the ‘.’ when running a program, they will be added by the software. If a directory is made manually, the ‘.’ must be put in. It is assumed that a database is always present in the system. The name of the default database is given by an environmental variable (see section 3.1), however if not set, it will default to AGA for agency. Here, BER will be used as an example throughout the manual. A database has a duplicate storage of the events. For quick reference and interactive work the events are stored in single files (S-files) in yearly directories and monthly subdirectories. When new data is entered into the database, it comes in as individual event files. However, once the interactive work has finished, the single event files are overwritten with the final location and additionally stored in monthly files, which are only changed when updating (UPDATE command, see section 14). The monthly files, called CAT-files for catalog, are stored separately in the CAT directory and primarily used for quick searching and backup for the single files. In addition to the event data, there is also a LOG directory in each database to keep a log of the data processing, see section 14.

#### S-file database structure

The structure for the single file storage is as follows (Windows example):

```
\REA\BER\_ Main readings directory, all data
\REA\BER\_1999\ Data for 1999
\REA\BER\_1999\01\ Data for January 1999, each event in one file
```

On Unix, the last line would have been `/REA/BER_/1999/01`

The structure works back year 0000. Each event contains original phase readings in the Nordic format (Appendix A) which includes file names of all corresponding waveform files. One event is one file. Each event has an ID line. The ID line contains a unique ID, which will follow the event through all COLLECT and SPLIT operations (see section 12 and 13). The ID is directly correlated with the S-file name and after an update of the S-file, also with the origin time. The ID line also contains status information about the event like last action, when it was updated etc. The ID-number is normally fixed, which is useful if data is taken out from the database, processed on another computer and later put back into the database, since otherwise the ID of an event might be changed and the existing file would not be overwritten. An example of an S-file name is :

```
27-1112-11L_S199401
```

The S-files are used as input for the location program and, when making a permanent update, also for output, see 7. The letter in front of the “.” indicates the event type and can be L, R or D for local, regional or distant event respectively. It is the same indicator as given in the header line of the S-file, see the Nordic format page 537. The remaining numbers give (in order) day, hr, min, sec, year and month.

As mentioned above, the system can contain many other databases, which may function exactly like the BER directory. A data base can be used to store a subset of data or data from different networks. Data can be moved between databases or in and out of the databases, for details, see description on EEV 5.3 and 5.4.
CHAPTER 2. STRUCTURE OF SEISAN

Monthly location files, the CAT directory

Events located in monthly files are in a directory called /SEISMO/REA/BER../CAT in addition to the individual S-files. Additional databases like e.g. NAO will have epicenters stored under /SEISMO/REA/NAO../CAT. The monthly epicenter files are called 199901.CAT for e.g. January 1999. Although the files generated by SEISAN normally are monthly files, the CAT directory can also contain yearly files or any other time interval. The only rule is that the name of the file must give the year and month of the first event in the file. This is because the search program SELECT can use the CAT files to search requested time intervals (SELECT most often use the S-files for selection). If a user has a historical catalog, this can be added as an individual file. If the historical catalog starts in 1820, the file name would be 182001.CAT. The files in CAT do not need to be continuous in time, but they must not have overlaps in time and each file must have data in chronological order. The format of the CAT files is the same as for the S-files. Additionally, CAT files can also be compact files, meaning just the header lines of the S-files (see also section 2.3).

2.2.2 Waveform data and formats

SEISAN works with various waveform formats including SEISAN, GSE2.0, SEED/MINISEED, GURALP gcf(single channel files), Helmberger format and SAC binary and SAC ASCII. The SEISAN format is described in Appendix 3 while for a format description of GSE and SAC the user is referred to GSETT-3 [1997] and Goldstein [1999], respectively. The SEED format is described in IRIS Consortium [1993]. The GSE reading routines are based on the codeco routines written by Urs Kradolfer, Klaus Stammler and Karl Koch. The routines read GSE2.0 only, not GSE2.1. The format description of GSE2.0 is given in: http://www.seismo.ethz.ch/prod/autodrm/manual/CRP243.0psAnx3.A4.pdf. The different formats can be used in parallel by several programs. With MULPLT for example it is possible to plot data in the four formats at the same time. Other formats can be added by adding reading routines and adding the respective calls to LIB/wave.for. Note that SAC binary files can also be used on Windows from SEISAN version 8.2. To use other formats, a conversion program must be used first, see section 19.

In general it is recommended to keep the waveform data in one format only, mainly for simplicity and maintenance reasons. There may be different arguments for or against one or the other format depending on the user’s preferences and requirements. SAC and GSE are widely used formats and therefore may be attractive. SEISAN is a multi-trace binary format with direct read access to individual traces. SAC is a single trace binary or ASCII format with a large number of header parameters. The SAC format is widely used in research-oriented programs. GSE is a multi-trace ASCII waveform format that includes various sub-formats. It is widely used for data exchange. Although the GSE format can keep any number of traces, it is recommended to include no more than 3 traces in a single file depending on the number of samples, since when reading a particular trace, the whole file may have to be read.

The MINISEED format is probably the best option since most data centers use it and almost all instrument manufactures will provide it. However, the SEISAN implementation should probably be tested a bit more. SEISAN cannot read SEED files using all options possible in SEED, but data from the largest data centers as well as many observatories have been used for testing. With respect to MINISEED, there are less problems since MINISEED is simpler than SEED. SEISAN can also write MININSEED (program WAVETOOL), but cannot write SEED (unless GSE2SEED is used). The WAV directory contains files with digital waveform data. The directory normally has no subdirectories or any other organization. However, in case of large databases, WAV can be subdivided, see below. In addition any directory can contain waveform data, it has to be specified in SEISAN.DEF (section 3.12).

The amount of data that can be stored is only limited by the disk size. The analysis system will always look in WAV for particular files if they are not in the user’s own directory. Waveform files will
automatically be transferred to WAV on initial registration into the database (see MULPLT). Registration is the process of automatically creating an S-file in the database with the name of the waveform file and header information. Phase pickings are done later. See section 8.

There is normally no requirement for particular filenames for the waveform files in WAV or elsewhere, however many programs will make file names like:

```
1995-01-23-1230-20T.BERGE_013
```

With the abbreviations yyyy: year, mm: month, dd: day, hh: hour, mm: minute, ss: second, T: file type indicator (normally S), NETWO: maximum 5 letter network code and nnn: number of channels.

Recommended file type indicators are: S: Standard SEISAN, R: Resampled, A: Appended, M: Miniseed/SEED

WAV database: In case a large number of waveform data is stored, it might be an advantage to also split up the WAV directory in subdirectories. This is done in the same way as in the REA directory, e.g. waveform files for BER from July 1994 would be found in WAV/BER_/1994/07. Programs that use waveform files will automatically search, in order, the current directory, TMP, WAV and the monthly WAV directory (TMP. see compression below). However, it is a requirement for all programs running outside EEV that the waveform data is in the default database since only that one is searched. When storing in the WAV database, it is a requirement that the waveform names by default start with either yyymm (like 9902) ,yyymmd (like 19990101) or yyyy-mm (like 1999-02). If this is not the case, the position in the file name of year (including century) and month must be specified in SEISAN.DEF, see parameter CONT_YEAR_MONTH_POSITION_FILE. In this case all the waveform files in the WAV structure must have the same type file name.

Waveform files created on Windows and Linux SEISAN version 7 or newer cannot be read on older SEISAN versions.

The SEISAN binary waveform format is explained in Appendix B. The files are written and read with the same Fortran statements on all platforms, however the internal structure and byte order are different. As of SEISAN version 5.1, files written on either machine can be read on the other and there is no need for any conversion when the binary waveform files are moved between Sun, Linux, MacOSX and Windows.

**Compression of waveform data**

Waveform files can be stored in compressed format. The compression must be done by the user. Programs that access the compressed waveform files copy the file to the TMP directory, and uncompressed there. The uncompressed file remains afterwards and will be found the next time one of the programs is looking for the same waveform file. The content of the TMP directory has to be deleted manually. On Unix, you may automatically delete the content of the TMP directory by a cronjob, see manual pages on crontab. On Unix the compression formats supported include gzip, compress, bzip2 and zip. On Windows, only gzip is supported (gz files). In order for it to work, the command gunzip must working. This can be done with gzip which also can decompress. First install gzip if not there. Then create a bat file to be in you part (like in COM) with name gunzip.bat and content

```
gzip -d
```

With the introduction of SEED format, there is less need for external compression since the SEED data usually is compressed and therefore decompressed on the fly when read. Also now disk is usually not a problem when dealing with individual events.

**Component codes**

The SEISAN waveform format until version 8.2 has used 4 characters for the component code. The
first character indicates the type of sensor, for example ‘B’ for broadband, ‘S’ for short-period or ‘L’ for long-period. For acceleration data the first character has to be ‘A’ because SEISAN assumes that the corresponding response has been given as acceleration response. The fourth character has to give the channel orientation, ‘Z’ is used for vertical, ‘E’ for east-west and ‘N’ for north-south. Other orientation of the horizontal components is possible in GSE, SEEK and SEISAN are not understood by SEISAN. If data are rotated, ‘T’ is used for transverse and ‘R’ for radial. The second and third characters can be chosen by the user. From SEISAN version 8.2, only 3 characters are used, the first 2 and the last. These 3 characters are then defined according to the SEED standard. SEED location codes and network codes are now also stored in the SEISAN format and are displayed when plotting the traces with MULPLT. From SEISAN version 12, several program will use location and network codes, particularly if the new Nordic Format is used. The component code is part of the response filename and is used to find the response corresponding to a given station and component. The network code is not part of the response files except for the SEED response files so if using a SEED response file, network and location codes must match. Program WAVFIX can be used to change station and/or component codes for most formata (but not yet network and location codes).

The Nordic format only has space for two characters for the component code. The definition in SEISAN is that these are the first and fourth character of the waveform component code. This means that the relation between the component code in the Nordic file and the waveform data is non-unique. In the New Nordic Format, there is room for 3 char component code and location and network codes. D. The GSE and SEED waveform formats have three characters for the channel code, see [GSETT-3 1997] and [IRIS Consortium 1993] for the detailed definition of the component codes. SEISAN, when reading waveform data in either GSE or SEED format internally keeps the first two characters and moves the third to fourth, so for example ‘BHZ’ becomes ‘BH Z’, however the user will only see the name as BHZ in many programs. Data files in SEED also have a location code, which allows to distinguish for example between two ‘BHZ’ components (for example a 30 second and 120 second sensor with the same sampling rate and high gain) at the same site. Z. When converting between SEISAN and SEED/MiniSEED, station, network and location codes are preserved while SAC and GSE only partly can store this information. SAC has more than four characters for the component code and sacsei.def parameter file in DAT has to be used to define the conversion. However, normally SAC data will have three character component codes as well. Conversion of component codes from SEISAN to SAC is also defined in sacsei.def.

When converting between SEISAN and other waveform formats, component conversion is defined in the respective definition files, see section on conversion programs.

2.2.3 Continuous waveform data

In SEISAN one can plot or extract continuous data from either a standard SEISAN database (internally many different formats) or from a BUD or a SeiscomP archive (internal format MiniSeed. The archive option is now the most common way to use continuous data in SEISAN, it is now easy to convert most format to MiniSeed with WAVETOOL and the use any group of MiniSeed files to create the archive with program DATATASELECT (included in SUP), see section ‘getting data into the data base’ See also page 76.

Continuous data in a BUD or SeiscomP archive

SEISAN reading BUD and SeisComP archives

We have implemented archive reading in SEISAN. The reading routines using Chad Trabant software have been implemented by Ruben Luis. Reading continuous data: This works just like reading SEISAN continuous data, except there are no S-files, only the archive files. All the same functions are available:
2.2. THE DATABASE

Plotting, zooming and extracting segments and registering events.

Read archive data as an event from eev:

A reference to a segment is made in the s-file and it is treated as if it was a file. When a keyword for
archive (ARC) is found, the reading is directed to the archive instead of to a file. The archive reference
is e.g.

```
ARC STAT COM NT LO YYYY MMDD HHMM SS DUR
ARC ROSA BHZ PM 2010 1011 0100 00 14400
```

where ARC indicate archive, STAT is station code, COM is component, LO is location code YYYY
MMDD HHMM SS is start time and DUR is duration in secs.

Thus the segment in archive with given start time and duration is considered a file. If later plots require
less data than the segment referenced, the whole segment is still read, like reading the whole trace in a
file in archive with given start time and duration. A mixture of archive references and file names can be
used.

The archive reference can also use wildcards to reference many channels, like e.g. just writing ARC and
the rest of the line blank, all channels in the archive will be selected.

Station is blank or *, component, network and location are blank: All channels defined in SEISAN.DEF
will be plotted with start time and duration given in ARC line (if not blank, see next option). If a
component is given, only that component will be selected. If a network is given, only that network will
be chosen. If a location code is given, only channels with that location code is selected.

Start time and or durations blank: Start time will be origin time - a time given in SEISAN.DEF
(ARC START), duration will be a time given in SEISAN.DEF (ARC DURATION).

Station is P: All channels for all stations listed in the S-file found in the archive will be plotted. There
is no requirement for the station to have any other information than the station name, component code
is not used. So a new station can easily be added.

Station begins with _ indicates virtual network as defined in SEISAN.DEF. This can also be inserted
with arc command in EEV.

Plotting all stations without an archive reference line: If parameter ARC BY DEFAULT in SEISAN.DEF
is set to 1, all channels in the archive will be selected.

NOTE: An ARC line can be inserted/edited in the S-file from EEV by command arc.

The archive is defined in SEISAN.DEF, where each channel is given by a ARC_CHAN text string: SSSSS-
CCCNNLL, where SSSSS is the station, CCC is the component, NN the network and LL the location.
These lines can be generated with program sample_read wav, which will make a list of all channels in
a given set of files (it thus requires to have one of several single files with the channels needed). The
directory of the archive is given by ARC_ARCHIVE, see e.g.:

```
ARC_CHAN PMOZ BHZPM
ARC_CHAN PMOZ BHNPM
ARC_CHAN PMOZ BHEPM
ARC_CHAN SFJD LHZIU10
ARC_CHAN SFJD LHIU10
ARC_CHAN SFJD LHEIU10
ARC_ARCHIVE /uibbsan/home/s2000/BUDARC
```
CHAPTER 2. STRUCTURE OF SEISAN

ARC_DURATION 10000.0
ARC_START_TIME 5000.0
ARC_TYPE 1.0
ARC_BY_DEFAULT 0.0

where each channel is defined as well as the location of the archive. The specification is the same for both BUD and SeisComp archives. ONLY one archive type can be used at the same time and the archive type, ARC_TYPE is given in SEISAN.DEF.

Groups of channels can be defined as virtual networks. The virtual network names will be shown in the station selection when using ‘arc’ option in MULPLT. The definition is done in SEISAN.DEF:

ARC_VIRTUAL_NETWORK _NO BER H3NS00
ARC_VIRTUAL_NETWORK _NO BER H3NS00
ARC_VIRTUAL_NETWORK _NO BER H3NS00

The virtual network name has to start with a ‘_’ to make them visible as such in the station selection.

Archive reading works on both Linux and Windows.

Continuous data in a SEISAN database

In SEISAN continuous data has no special format. Continuous data is simply ordinary waveform files that follow each other in time. In order to treat the data as continuous, the data can be put into a SEISAN continuous data base. Such a data base is made as follows:

- For each waveform file from a station or network, an S-file is created. The S-files only contain reference to the waveform file(s). Program AUTOREG can be used to create the S-files.
- The waveform data is optionally put into the corresponding waveform station directories, however they can also be in WAV or working directory. For large data sets it is strongly recommended to use the WAV database structure.
- The continuous databases are defined in SEISAN.DEF in DAT.

If e.g. data is to be stored from 3 different stations (three component files), create 3 databases under WAV and REA with the name of the stations (program MAKEREA). If the continuous data consist of 20-minute files, this would mean about 2200 files per month, which is a reasonable number. It is now possible for some programs (MULPLT, WAVETOOL) to get access to any or all of the traces in the continuous data base and plot and extract data. If the continuous data is archived from a real-time system it is best to have one database per station as it will at times be necessary to backfill gaps as data may not have arrived in real-time.

Note that waveform files in a SEISAN continuous structure must contain the year and month which is used to locate the corresponding year-month structure. The default start of the file names accepted are:

ccmm ccyy-mm ccyyymm

where cc is century, yy is year and mm is month. If year and month are placed differently in the file name, their location must be specified in SEISAN.DEF, parameter CONT_YEAR_MONTH_POSTION_FILE.

It is also possible to store the data without having a database for each station:

- Alternative 1: If the 3 stations have waveform files starting at about the same time and the same duration, they can be merged to 9 channel files and only one continuous data base is made. This
may work well for data from a temporary deployment where all data is there when the data is put into database.

- Alternative 2: If the 3 stations have waveform files starting at about the same time and the same duration, the 3 waveform files can be listed in the S-file and only one database is needed.

- Alternative 3: If the files are in individual channel files, 9 waveform files can be listed in the S-file and only one continuous database is needed.

The waveform files in a continuous database can have different formats for different stations and one S-file can refer to more than one waveform file, provided they start at about the same time and have the same duration.

A simpler way to use smaller quantities of continuous data is to make a list of these files with DIRF and an application program can then use that list to work with the data. Currently two programs have special options for this kind of continuous data. The MULPLT program will plot data from several files as if it was one file in one continues trace the RESAMP program will resample the data from several files and put it into one output file.

### 2.3 File types used with SEISAN

A description of the different file types is given below with typical names. Most names must be exactly as specified, others can be given names. However it is VERY important that no name including full path is more than 80 characters long. Until now this has not been a problem, however it has to be considered when SEISAN is installed.

It is extremely important that no ASCII files contain tabs, use blanks instead, a tab will screw up formatting. Similarly non standard ASCII characters cannot be used.

The basic unit is a file in the Nordic format, (see Appendix[1]). For practical purposes 3 descriptive names are used for Nordic files:

- **S-file**: Single event file with phase readings, with or without source parameters such as location and magnitude. In the database these files are named with the extension: .Syyymm This is the standard type of file in e.g. the BER/1998/08/. An example is 11-1234-11L.S199808.

- **CAT-file**: A catalog file containing many S-files with location or just a catalog of hypocenters, a compact file, see below. This is the standard type of file in e.g. the /REA/Ber/CAT directory. An example is 199801.CAT. This file format is also output from several programs like SELECT and COLLECT. There is a blank line between events.

- **Compact file**: This is a CAT-file with only the source information. One event is represented with one line, (the header line in the S-file). There is no blank line between events. A compact file can be generated by either COLLECT or NORHEAD (earlier called COMPACT).

In addition there are the following types of files:

- **SEISAN waveform file**: Waveform data can be stored in SEISAN, GSE, SEED, MiniSEED, Guralp, Helmberger and SAC format, see section 2.2.2. An example of a name is 1992-01-11-2233-22.S.Berge011.

- **Response file**: File giving the response of a given channel at a given station. They are typically generated with the RESP program, see description of CAL directory, section 4.4. This is the standard type of file
in the CAL directory. An example of a name is ODDA_S_Z. 1999-05-01-0000 SEI. However, SEED and SAC response files extracted with rdseed or made with other programs can also be used.

File listing: This is just a file with a list of numbered files. The file name is always filenr.lis, and it is generated with the DIRF program, see 15.

Index file: This file contains a listing of absolute paths to a series of S-files. The index file can be used as input instead of the CAT-files to several programs. Several programs generate index files as e.g. select and eev. The index file has the same format as the filenr.lis files described above and can be generated with the dirf command using S-files. The index file name must contain a ‘.’. An example is shown below:

1. \SEISMO\REA\TEST\1993\09\29-2228-26D.S199309
2. \SEISMO\REA\TEST\1994\06\16-1841-57D.S199406
3. \SEISMO\REA\TEST\1996\06\03-1955-40D.S199606

2.4 Upper and lower case

Upper and lower case file names only makes a difference on SUN, Linux and MacOSX. The intention is that all permanent data file names used by SEISAN should be in upper case (e.g. S-files, crustal model file, directories (e.g. REA) while temporary files should be in lower case (e.g. print.out). Programs are also in lower case. It should then be a bit more difficult to delete the permanent files. NOTE THAT THROUGHOUT THIS MANUAL, PROGRAM NAMES ARE GIVEN IN UPPER CASE TO INDICATE THAT THEY ARE NAMES, HOWEVER WHEN USING THE PROGRAMS, LOWER CASE MUST BE USED ON SUN. In program MULPLT, commands are case dependent.

2.5 Moving data between Sun, Linux, MacOSX and Windows

All S-files and file names are identical on the three platforms. To move many events (S-files) from one system to another, make a COLLECT (section 12) on the original system and a SPLIT (section 13) on the receiving system. As mentioned in section 2.3, the SEISAN binary waveform files have different internal structure if written on Sun, Linux, MacOSX or Windows, but this is corrected for in the reading routine, so files can be copied directly. GSE files can be copied directly since they are ASCII files. SEED/MINISEED files can be used directly on all platforms.

The only other files that are different are the binary earth model files IASP91.platform.HED and IASP91.platform.TBL in the DAT directory (where platform is either linux, macosx or windows). The platform is included in the filename. The files cannot be moved, but are easily regenerated with the IASP91 programs, see section 30 and 8.8 in the Hypocenter manual.
Chapter 3

Installation

SEISAN has been tested and compiled for Windows (/Vista/7/8/10), Linux and MacOSX. Earlier Solaris was also supported. The SEISAN GUI “Seisan Exporer” is provided for Windows, it must be compiled for Linux and it has not been tested for MacOSX.

Upgrade from previous versions

Before you start, take a backup copy of your DAT directory. Note that when you upgrade, many parameter files will be overwritten so make sure old parameter files are copied before putting in a new version of SEISAN. The most important are in DAT: STATION0.HYP, SEISAN.DEF, MULPLT.DEF. Also the Unix setup file SEISAN.csh and SEISAN.bash is overwritten. You may also want to keep copies of PRO, LIB and INC to keep a copy of the old source code, especially if you have done any modifications to the code. You can keep almost all of your parameter files, only SEISAN.DEF has been changed. Check this file and change to your system. Some individual program parameter files like for SPEC have changed.

How to get SEISAN

SEISAN can be downloaded using a browser from


or on command line with

ftp ftp.geo.uib.no

login is ftp and password is your email address. On the AFTP server go to /pub/seismo/SOFTWARE/SEISAN. Use binary mode for the compressed files (tar and zip). Before copying, check the readme file for latest updates, changes and current content of the directory. The directory will at least contain the following files:

| seisan\_X.Y\_unix.tar.gz | a compressed tar file, whole distribution with executables and test data, X.Y stands for the latest distribution number and Unix for the respective Unix system (linux or macosx). |
| seisan\_\_X.Y.exe | Windows distribution an install file |
| seisan\_X.Y.pdf | The SEISAN manual, Adobe PDF |
| seitrain\_X.Y.pdf | SEISAN training course |
| testdata\_X.Y.tar.gz | SEISAN data for the training course |

Section 3.10 gives additional information about modifications and recompilation.
3.1 Linux

Linux: The programs in the distribution have been compiled under Red Hat Enterprise Linux using the GNU compilers gcc and gfortran. It is recommended to recompile the programs, since otherwise the programs might not run on your Linux distribution. In the Redhat distribution of Linux the Fortran compiler is not part of the standard distribution, it has to be installed (see your Linux manual for instructions). THE USER ACCOUNT MUST BE SET UP TO USE csh, tcsh (use SEISAN.csh) or bash (use SEISAN.bash), in order for the SEISAN scripts to work. Note that in the following SEISAN.csh stands for both SEISAN.csh and SEISAN.bash. Otherwise the scripts need to be adopted to the shell used.

Instructions
The first step is to install the distribution, the procedure is the same for all Unix platforms.
1. Get tar.gz file
Copy the distribution file for your platform through FTP or from the web site to the SEISAN top directory, this could be a directory seismo under the home directory.
2. Decompress
   gunzip seisan_version_system.tar.gz
There should now be the uncompressed file in your directory (without .gz).
3. Install SEISAN
   tar xvf seisan_version_system.tar
Check that the SEISAN directories have been created.

If SEISAN has been installed without executable files, they can all be generated with the command ‘make all’ from the PRO directory. On Linux/MacOSX it requires the GNU Fortran compiler gfortran. See also section on compilation [3.10].
4. Regenerate the IASP travel time tables
On at 64 bit computer the IASP files in DAT must be regenerated if you have the files from a 32 bit computer, with the programs REMODL and SETBRN otherwise HYP will crash. First run REMODL, then SETBRN. The program do not require any input.
5. Set system parameters
If you are doing an update, some of the following settings can be skipped.

Activate SEISAN:
csh/tcsh shell:
In your .cshrc file, the aliases and paths used by SEISAN are defined by adding the line
   source /home/seismo/COM/SEISAN.csh
where ../seismo is the directory below which SEISAN has been installed. The SEISAN.csh script file assumes that you are running either csh or tcsh as your shell.
bash shell:
If you are using the bash shell add this line to your .bashrc file:
   . /home/seismo/COM/SEISAN.bash
bash might include a select system program, if that is the case on your pc you also need to add this line in your .bashrc file:
   alias select="/home/seismo/PRO/select"
3.1. LINUX

to use the SEISAN SELECT program.
If you are using another shell you need to modify the script accordingly or change the shell. It is assumed
that X-windows is installed.

SEISAN path for programs:
In order for programs and subroutines to know the path to the SEISAN program directory, this must be
defined in the file SEISA.csh or SEISAN.bash in COM. Edit that file and set the environmental variable
SEISAN_TOP to the name of the top directory, meaning the directory structure below and including
seismo e.g. /top/users/seismo. This variable is then used to set the path to SEISAN directories.

SEISAN path for databases, parameter files etc:
The SEISAN database can be under the same top directory as programs, however it can also be different.
This is practical if several users have their own databases, but use the same software. Set environmental
variable SEISAN_TOP to top directory e.g. /top/users/seismo.

SEISAN agency:
In SEISAN.csh also set the environmental variable AGENCY (upper case) to your 3-letter agency code
(upper case). This variable is only used by program MACROIN from EEV in connection with entering
macroseismic data so for most users ignore this setting.

SEISAN default database:
To locate the default database directory set environmental variable DEF_BASE in SEISAN.csh. If not
set, the name TEST is used since the distribution comes with some test data already installed in TEST.
The data bases are found under SEISAN_TOP.

SEISAN editor used in EEV:
The default editor is vi, any other editor can be set with the environmental variable SEISAN_EDITOR.

SEISAN calibration file directory:
By default, calibration files are in CAL, but they can be in a directory set with variable LOCAL_CAL.
The directory name must be complete like /home/users/calibration/

SEISARCH

Gives the architecture, can be either linux32, linux64, macosx or windows. Used in Makefile when
compiling.

SACLIB
Specify path and filename to SAC libraries, only needed when you compile programs (Unix) and you
have the libraries installed on your system.

Printer for Postscript plots:
The hard copy files from programs are sent to the printer from within the programs using the standard
lpr command. In the SEISAN.csh file, define lpr using the standard environmental variable PRINTER.
Remember that the printer must accept Postscript. PostScript files can also be viewed and printed on
most printers outside SEISAN using GhostView, however in that cases files cannot be printed from within
a program.

Scaling for Postscript plots:
By default, plots will be in A4 size. This can be changed by setting the environmental variables
SEISAN_PSCALE_X and SEISAN_PSCALE_Y. The default for A4 size is 1.0 for both variables. For
Letter size the Y-scaling can be set to 0.9.

Seisan Extension:
User specific code can be implemented by making use of the environmental variable SEISAN_EXTENSION.
CHAPTER 3. INSTALLATION

The idea is that programs read this variable, if set to the user specific string, the user’s source code will be used instead of the default. An example could be the computation of error ellipses. Currently used codes are: BGS.

6. Testdata
The testdata set can be extracted from the file `testdata.X.Y.tar.Z`. Use programs uncompress and tar to extract the data in the SEISAN top directory (keep subdirectory structure). SEISAN has 2 test events always installed with the software (from June 1996) so it is always possible to test SEISAN without installing test data. See the SEISAN tutorial, which only uses the 2 test events.

**Dimensions**

Most dimensions are set in file `seidim.inc` in the INC directory. In order to change dimensions, first change in the include file and then recompile the whole SEISAN distribution. The most important dimensions are:

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of points in one trace</td>
<td>2 400 000</td>
</tr>
<tr>
<td>Number of points in memory buffer</td>
<td>30 000 000</td>
</tr>
<tr>
<td>Number of lines in NORDIC format file</td>
<td>20 000</td>
</tr>
<tr>
<td>Maximum number of traces in one plot</td>
<td>1 800</td>
</tr>
<tr>
<td>Maximum number of events in one month</td>
<td>200 000</td>
</tr>
<tr>
<td>Maximum number of calibration files</td>
<td>4 000</td>
</tr>
<tr>
<td>Maximum number of epicenters in epimap</td>
<td>90 000</td>
</tr>
<tr>
<td>Maximum number of lines in index file made with dirf</td>
<td>200000</td>
</tr>
</tbody>
</table>

SEISAN has been tested with much larger dimensions, like 10 000 000 for number of points in one trace, however large dimensions might slow down the speed due to swapping (particularly if memory is not large) so a smaller dimension has been chosen. For continuous data, SEISAN works with many files so smaller dimensions can be used. For the PC version, dimensions may be different from above, check `seidim.inc`.

**Note:** In case programs don’t work, you might have to recompile, see section [3.10](#).

**Ubuntu 16.04:** you might need a extra library to run SEISAN, run: `sudo apt-get install libgfortran3`

Some Ubuntu users are missing the `libg2c.so.0` library file, it can be installed with the command (you might need to be online):

`sudo apt-get install gcc`

If this does not work, also try:

`sudo apt-get install libg2c0`

On a 64 bit computer the IASP files in DAT must be regenerated if you have the files from a 32 bit computer, with the programs `REMODL` and `SETBRN` otherwise HYP will crash.

**Multiple users on Linux/Unix**

If two or more users are working with EEV at the same time with the same user, there is a risk that the S-file names are being mixed up so an event from one year suddenly gets the S-file name from another year. This is caused by both users using the same environmental variable for the S-file (should be changed!). The solution is that each users has his/her own account, which in any case is the most convenient.

From version 12 there is a lock file so if two users use EEV in the same directory, a warning is given. More details under EEV.
3.2 MacOSX

The MacOSX version does not come pre-compiled, and will have to be compiled by the user. Please see section 3.10 and 3.10.2 for details.

3.3 Cygwin

As alternative to running SEISAN directly under Windows, it can also be compiled and used under Cygwin. The cygwin website [http://www.cygwin.com](http://www.cygwin.com) explains: “Cygwin is a Linux-like environment for Windows. It consists of two parts:

- A DLL (cygwin1.dll) which acts as a Linux API emulation layer providing substantial Linux API functionality.
- A collection of tools which provide Linux look and feel. The Cygwin DLL currently works with all recent, commercially released x86 32 bit and 64 bit versions of Windows, with the exception of Windows CE.”

It is attractive to use SEISAN with Cygwin as it looks like the Unix version. Under Cygwin SEISAN uses X graphics, which requires the Cygwin X server (part of Cygwin). The compilation under Cygwin is like Linux, which means on the software side there are fewer differences to take care of.

Cygwin can be downloaded and installed from the website. However, to be able to compile SEISAN a number of non-default packages have to be included:

- Devel: gcc-g++ (and associated packages)
- Devel: gfortran
- Devel: make
- Libs: libncurses-devel
- Shells: tcsh
- X11: xorg-x11-base
- X11: xorg-x11-devel

We also recommend to install:

- Editors: vim
- Utils: diffutils

The choice of packages is done through the Cygwin installation tool. With some of the packages, additional required packages will be selected by default and must be installed.

To install SEISAN under Cygwin, use the Linux distribution and set SEISARCH to ‘linux32’ or ‘linux64’. You have to compile as described for Linux below.
CHAPTER 3. INSTALLATION

Under Cygwin, you can use csh or tcsh, which will allow to use the SEISAN.csh. If your default is bash, ‘(t)csh’ can be started from the command line. The X server is started using command ‘startx’ from the Cygwin prompt. See the Cygwin documentation for more details.

Cygwin has not been tested on latest versions of SEISAN.

3.4 Windows

It is assumed that you transferred the Windows distribution from Internet to a directory on the PC. In the following, it is assumed that you install on disk drive C. The windows graphics work with any resolution, small fonts should be used, however, SEISAN is designed to work with 1024x768 or more.

SEISAN has not been tested on Windows XP or earlier Windows versions. Vista and W8 has problems with some graphics programs. We recommend to use W10.

Using install script

Click on seisanwindows.msi and follow the instructions, use all defaults. If you later want to change some environmental variables, see instructions below. If SEISAN already is installed, a window comes up for a possible upgrade or removal of SEISAN. A new version cannot be installed without removing the old one. SEISAN is now completely installed. Optionally, some parameters can be set, see below. The general non system specific parameters are the same as under Linux.

OPTIONAL: Change environmental variables SEISAN_EDITOR, SEISAN_PSSCALE_X, SEISAN_PSSCALE_Y and DEF_BASE (see UNIX section for definition) using control panel - system - advanced and select environmental variables. The defaults are respectively

SEISMO, notepad, A4 and TEST. SEISAN_TOP would be set as set SEISAN_TOP=\SEISMO, but could also be e.g. test\best\analysis or d:\seisan. Note there is one blank character at the end and the first character MUST be “\\” or the second “:”. If no seismo directory, it could e.g. be just E:

Optional: If you want to recompile, you need the Fortran Gnu compilers installed, see section on compilation below.

Printer:

It is assumed that a Postscript printer is connected to the PC, either directly or by a network. When a program sends a plot to the printer, it issues the command SEISANPR. In the COM directory, there is a file called SEISANPR.BAT containing the print command. Several suggestions are made in that file, the default is to use COPY filename PRN. If you do not have a PostScript printer, it is possible to view and print PostScript files with GhostView, a public domain software available from many sites, e.g. . Other software is also available to view PostScript files.

http://www.cs.wisc.edu/~ghost/gsview/get47.htm

Testdata:

The testdata set can be extracted from the file seisan_test_data.tar.gz with for example the WinZIP program. Extract the data to the SEISAN top directory. To use the test data base as the default DEF_BASE must be set to TEST (done by default during installation).

Running SEISAN on networked PC’s

SEISAN on one PC can be accessed from another PC. This can be an advantage if several users share the same data base. On the client PC, name the server PC disk where seisan is installed e.g. Z:. Then set seisan_top=Z:\. 

http://www.cs.wisc.edu/~ghost/gsview/get47.htm
3.5. **PYTHON**

Potential problems

Program takes a long time to load: If the program is large, it might use disk swap files, which can take a lot of time. Reduce array size (seidim.inc in INC) and recompile.

Commands like P or L do not work in EEV. If swapping takes place, some damaged swap files or other files might remain which prevents the program from starting. Clean disk with command SCANDISK.

**Graphics problem:** Graphics programs write text to a console window. If there is a program crash, the error message goes to the console window, which then disappears.

3.5 Python

Python is currently only used with two program in SEISAN, MOPAD (making moment tensor plots) and SPECTRO (making spectrograms) so if any of these program are to be used, Python must be installed. Most Linux systems will have Python installed, else follow standard instructions on Python home page and install. SEISAN currently uses version 2.7 of Python. Make sure you include Python in the path.

For running MOPAD, two extra libraries are needed. In Windows they are install like this:

```bash
pip install numpy
pip install matplotlib
```

For SPECTROGRAM, more libraries and Obspy are needed. On Windows this is done as follows:

```bash
pip install scipy
pip install lxml sqlalchemy
pip install pyreadline ipython
pip install flake8 nose mock
pip install obspy
```

In windows, Python files must be recognized as a Python programs so environmental variable PATHTEXT must have .PY if not already there.

On Linux, follow instruction on ObsPy home page https://github.com/obspy/ob1Gspy/wiki

3.6 Google Earth

GoogleEarth is currently used with the GMAP program and from EEV. It is also an option to plot automatically with Google Earth when locting from EEV or SE. Googel Earth can be downloaded from http://earth.google.com/download-earth.html. In order to use the automatic plotting in connection with location, Google Earth must be in the path. This is normally the case for Linux but not for Windows. check where googleEarth is installed, a typical location is c:\program files\google\google earth pro\client;

```bash
goto edit environmental variables
```

add the install location to the path
CHAPTER 3. INSTALLATION

3.7 Database directories for your own system, MAKEREA

When you want to use SEISAN with your own data, the necessary directory structure must first be created and your own station coordinates and crustal model must be defined. Both REA and WAV directory structures are created with the program MAKEREA. The program asks for database name (must be 1-5 letters and UPPER CASE), start time (year and month), end time (year and month) and the structure to create (REA or WAV). An output will then show the names of directories created. If a directory already exists, a message will be given. It is possible to create both structures at a time, since the program after finishing asks if the respective other structure should be created.

3.8 Default parameter files for the main programs

Once you start locating events, the location program HYP will look for an input file with station coordinates, location parameters and crustal model. This file is located in the DAT directory and called STATION0.HYP. To edit the file, move to the DAT directory by using command DA and edit the file. For more information about this file, see section 7. If you want to keep the original file for test purposes, first copy it to a file with another name.

When plotting epicenters (command MAP (no questions) or EPIMAP (user specify)) input files with map-contours for the EPIMAP program are used (type *.MAP). These files are also located in the DAT directory. If you want to use more detailed map contours, you must get hold of your own data and put them into file called e.g. MYMAP.MAP. In the DAT directory there are two sets of contours: WORLD.MAP and EUROPR.MAP. Detailed map files in SEISAN format for the whole world is found at the SEISAN ftp site.

The plotting program MULPLT can use a default file for those stations, which are to be routinely plotted, as well as other default information. This is defined in MULPLT.DEF in DAT, see example file. MULPLT.DEF also defines which keys are assigned to which phases and what character is used for the mouse. An example is given in DAT, see also example in 8.

Both the MULPLT.DEF and STATION0.HYP can also be in the working directory. Programs always look there first and on a multi-user system, this enables different users to have their own setup. It also makes it possible to work with different setups by just changing directory.

The AUTOPIC PROGRAM requires AUTOPIC.INP in the DAT directory.

Most SEISAN program use the SEISAN.DEF file (in DAT, see section 3.12) where many general parameters are set.

The bulletin program requires a front page with whatever you like and some set for fonts can be made. The text of this page is located in the BUL.INP file in the DAT directory.

3.9 Color settings

All programs using color can use a color definition file called COLOR.DEF. The file can be located in the current working directory or in DAT. Programs will first look in the working directory, then in DAT. If no COLOR.DEF file is given, default is used. Be careful with color setting, unexpected results might occur, like getting a blank screen when plotting white traces on a white background. Several color settings are universal like setting colors for titles, however several color settings are specific for particular programs,
see example file below. Since colors can also be used for the Postscript file, there is an option for using
color on the screen but not for Postscript, since it is more likely that the user will have a color screen
than a color Postscript plotter. Sending a color plot file to a black and white laser printer results in a
plot with gray tones.

The COLOR.DEF file:

This file is for defining SEISAN color objects and for setting the size(of full screen) of the graphics window
for individual programs. If no file available, default colors, as given at end of line below, will be used.
The entries can come in any order, however the keyword MUST appear as shown below. If an object
color is not defined, it will retain its default value. If a black or white background and no colors are
chosen, all colors will be reset accordingly.

Since version 9.0, the size of the window (% of full screen) for most graphics
programs is also set in this file.

The color codes are:

1: blue 2: green 3: red 4: yellow 5: white 6: black

color_screen 1 ! 0: no colors, 1: colors
color_hard_copy 0 ! ---------------------
color_back 5 ! background color
color_trace 6 ! seismic traces, map contours
color_pic 3 ! phase picking
color_zoom 2 ! zoo lines in mulplt
color_def 6 ! default color
color_frame 1 ! frames like epimap map frames, mulplt
color_title 6 ! titles on top of plots
color_spec 1 ! spectras
color_axis_not 3 ! axis notations
color_epi 3 ! epicenters
color_station 3 ! seismic stations
color_map_contour 1 ! epimap contours
color_map_grid 6 ! Lat-long/(x,y) grid
color_label_grid 6 ! Grid labels for map
color_symbol_key 6 ! Diagram key.
color_prompt 1 ! Prompt text.
color_section 3 ! section outline in epimap
color_bval_np 2 ! bvalue, number of events
color_bval_ac 1 ! ------, accumulated ---
color_bval_line 1 ! ------, lsq line
color_box 5 ! box for interactive input
color_box_letter 6 ! letters in ------------
color_foc_dilat 3 ! focmec dilatation
color_foc_comp 1 ! focmec compression
3.10 Compiling SEISAN programs

This section describes how to compile SEISAN or individual SEISAN programs. In the following subsections guidelines for individual operating systems are given.

The SEISAN distribution for all platforms includes the executables. Therefore in general it is often not necessary to recompile. In the case of Linux, it is often necessary to recompile since since different flavors of Linux behave differently. However, you may have the source distribution or you might want to modify some of the programs for your own needs or remove bugs and will have to compile programs.

The SEISAN programs on all platforms can be compiled using the `make` utility. On all platforms there is a `Makefile` in both the `PRO` and `LIB` directories and the `Makefile` is the same for all operating systems supported. The file might not need any modification, however the following parameters must be set correctly:

`SEISARCH` (environmental variable): This variable is used as keyword for the compilation, and can be `linux32`, `linux64`, `macosx`, `macosxppc` or `windows`. While the gfortran option should work on all platforms, the other keywords allow to have specific compile options. The keywords are also used to define which programs are compiled and installed in addition to the default list of programs. See chapter 7 for differences between the platforms. *(Note: Without setting `SEISARCH`, the compilation will not work since `make` will not know what `SEISARCH` is). On Linux/Unix system `SEISARCH` is set in the files `SEISAN.csh` or `SEISAN.bash` in `COM` while on Windows it is set in the `Makefile` itself or it can be set manually as an environmental variable. Most of the programs are the same on all platforms, but not all.

The compilation can now be started from the `PRO` directory by starting `make all`. From the `Makefile` in the `PRO` directory, the `Makefile` in the `LIB` directory is started to create the object libraries. A SEISAN archive in `LIB` for SEISAN routines is created, `seisan.a`. If you want to use SEISAN subroutines in your own programs you can link to that. Finally all programs are compiled.

Single programs can be recompiled by starting `make program`. If you do changes in the `LIB` directory
you need to compile using `make all`, which will also create the archive file. Then you can recompile individual programs in PRO as explained above.

Note that on all platforms the Chad Trabant MiniSeed library is used (new from version 9.0) and it is put into seisan.a. In the distribution they are located in file libmseed.c in LIB and 4 include files in INC (see seisan.all in INF). The libmseed.c file contains all subroutines in the original Trabant distribution and all include files from Trabant distribution are in INC. The current version of the Trabant distribution is 2.6.1. If you want to use different version, the same process as described above must be done.

**Seisan Explorer**

The compilation of the SEISAN GUI Seisan Explorer is described in the following subsections.

The source code se-source.tar.gz is found in SUP directory.

**IASP91**

If the computation of IASP91 traveltimes does not work after compilation, first run program REMODL, then program SETBRN to fix the files.

### 3.10.1 Linux

Compilers used for SEISAN (other version may also work):

- Linux 64 bit: Gfortran
- Linux 32 bit: Gfortran

Compiler installation:

For Linux/Unix, compilers are usually installed when the operating system is installed. If graphics programs do no link on Linux/Unix systems, check that you have X11 libraries in

```
$LD_LIBRARY_PATH
```

You can check what is there by command

```
echo $LD_LIBRARY_PATH
```

**SUSE**

When compiling SEISAN on SUSE Linux it has been suggested to use the following compiler options `\ `-malign-double -finit-local-zero'` in addition to the ones already used. Testing this on Redhat Linux produced very large executables, but it may be worth trying on SUSE.

**Ubuntu**

On Ubuntu one might need to add these libraries in order to compile SEISAN:

```
sudo apt-get install gfortran
sudo apt-get install libgfortran3
sudo apt-get install libx11-dev
```

**Seisan Explorer**

The source code se-source.tar.gz is found in SUP directory.

The GUI Seisan Explorer can be compiled on Linux. SEISAN must be fully installed on the computer before Seisan Explorer can be compiled. To compile Seisan Explorer on a Linux system, the Qt 5 framework is needed. This can be downloaded from https://www.qt.io/developers (if not already installed on your system). The qmake program is a part of the Qt framework. You need qmake in your path. Alternatively, you can specify full path to qmake.

Building Seisan Explorer:

1. Move into the directory containing the SE source code.
2: Copy the SEISAN library seisan.a from $SEISAN\_TOP/LIB to the linux subfolder.
3: Run `qmake se.pro` to build the makefile.
4: Run `make` to build the executable named `se`.
5: Copy the `se` to the `$SEISAN\_TOP/PRO` folder.

Step 3 and 4 is not needed if Qt Creator is used.

### 3.10.2 MacOSX

The compilation of SEISAN on a Mac is basically the same as for Solaris/Linux, in `COM/SEISAN.bash` or `COM/SEISAN.csh`, set SEISARCH to 'macosx' (Intel-based=newer Macs) or 'macosxppc' (PowerPC based).

You may also need to change the line `$\{fc\} seed.for` to `$\{fc\} -fno-range-check seed.for` in `LIB/Makefile`.

If you have gcc/gfortran 4+ installed and your Mac is Intel-based, you should now be able to compile. You also need X-windows, which should be preinstalled or on the installation disk for OSX 10.5 and higher (for earlier versions, they can be downloaded and installed).

New Mac computers are 64 bit, this URL is also showing how to tell if you Intel based is 32 bit: [http://support.apple.com/kb/ht3696](http://support.apple.com/kb/ht3696)

#### Additional hints on MacOSX 10

In order to compile and link Seisan off the source distribution, you need to have gcc/gfortran and Xcode installed.

We recommend you install the following:

- **brew package manager:** Copy the following line in the Terminal and press Enter: `/usr/bin/ruby -e "$\{curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install\}"
- **gcc compilers:** Upon successful installation of brew, execute "brew install gcc" from the terminal.
- **XCODE:** Either install it from the Mac store under
  

  **X11 Quartz:** Download the dmg-file, doubleclick and install:


The compilation procedure is the same as for Linux. However, for more details see INF/readme_macosx.txt.

**Seisan Explorer**
The GUI Seisan Explorer have not been tested on Mac!

### 3.10.3 Windows

Compiler installation: For Windows, the gcc/gfortran compiler is found at [http://sourceforge.net/projects/mingw/](http://sourceforge.net/projects/mingw/) The Fortran compiler and the MinGW development toolkit must be installed. SEISAN
can be compiled using the GW shell where the path to compilers is known. If you want to compile outside
the GW shell (in the DOS window) in the same way as under Linux/Unix, the compilers must be defined
outside the shell by adding c:\mingw\bin and c:\mingw\msys\1.0\bin to the path (assuming MinGW
installation under c:\mingw). Make sure path to SEISAN COM and PRO directoris come before path
to MinGW since MinGW has some commands identical to SEISAN like pr and test. Putting in the path
to MinGW also has the advantage of getting Linux commands in the DOS window like ls, tar, grep etc.
For the Windows platform, a graphics library and an include file is used from the DISLIN software
(new from version 9.0). Files dislin.h and dismg.a are located in INC and LIB, respectively. See [http://www.dislin.de/](http://www.dislin.de/) for more information.
Compilers used for SEISAN:
Windows: Gfortran under MinGW

Seisan Explorer

The GUI Seisan Explorer can be compiled on Windows. SEISAN must be fully installed on the computer
before Seisan Explorer can be compiled. To compile Seisan Explorer on a Windows system, the Qt 5
framework is needed. This can be downloaded from https://www.qt.io/developers. The qmake program
is a part of the Qt framework. You need qmake in your path. Alternatively, you can specify full path to
qmake.

Building Seisan Explorer:
1: Move into the directory containing the SE source code.
2: Copy the SEISAN library seisan.a from LIB to the windows subfolder.
3: Run qmake se.pro to build the makefile.
4: Run make to build the executable named se.
5: Copy the files se to the PRO folder.
6. Create a shortcut to ‘se.exe’ on your desktop if needed.
Step 3 and 4 is not needed if Qt Creator is used.

3.11 Program Version

The version number of this SEISAN distribution is given on the front page of this manual. New from
version 7.0 is that all individual programs have a version number and a date of the last changes. This
will help the users and developers to find out which version is in use and from which date. This is
particular useful when bugs are reported. In case major bugs have been fixed, the new versions (same
version number, different date) of the individual program will be available on the anonymous ftp site and
a message will be sent to the seisan mailing list.
In all programs you can obtain the version number by starting a program with ‘–version’ as argument.
For example collect –version will show you the version and date of your COLLECT program.

3.12 Setting general SEISAN defaults, SEISAN.DEF

There are several parameters which are common for several programs which are set in the file SEISAN.DEF.
This file can be in your local directory or the DAT directory. The system will first look in the local
directory, but normally the file is in the DAT directory. The file does not have to be there since there
are defaults for all parameters, but the parameters might need to be set.

The format of the file is as shown below:

This file is for defaults for SEISAN and called SEISAN.DEF. The name must
be in upper case on Linux. The following shows the parameters which can be set.
The file can contain any lines in any order, only the lines with
recognized keywords and a non blank field under Par 1 will be read. The
comments have no importance.

**********NEVER USE TABS IN THIS FILE, IT WILL NOT WORK***************

KEYWORD...........Comments.............Par 1.....Par 2
#
# waveform files location and other waveform stuff
#
WAVEFORM_BASE Waveform bases
WAVEFORM_DIRS Waveform directory /net/seismo/seismo/WOR/seisnet
MERGE_WAVEFORM Code for merging wa NSN
COPY_WAV_DIR Copy when register
CONFIRMATION Confirm when delete
#
# seisan cont data base
#
CONT_BASE REA continuous base RUND
CONT_BASE REA continuous base JMI
CONT_BASE REA continuous base NSS
CONT_BEFORE start min before 20.
CONT_AFTER start min after 1.
#
# position in file name for continuous files where year yyyy and month mm starts
#
CONT_YEAR_MONTH_POSTION_FILE
#
# archive
#
ARC_CHAN SFJD LHZIU10
ARC_CHAN KBS LHZE10
ARC_CHAN DAG LHGE
ARC_CHAN TULEGLHZDK
ARC_CHAN BSD LHZDK
ARC_CHAN BORG LHZII10 20020101 20150202
ARC_ARCHIVE ./archive
ARC_DURATION 10000.0
ARC_START_TIME 100.0
ARC_TYPE 1.0
ARC_BY_DEFAULT 0.0
#
# virtual networks
NAME SCNL
#
# several networks can be defined
3.12. SETTING GENERAL SEISAN DEFAULTS, SEISAN.DEF

# ARC_VIRTUAL_NETWORK _DK BSD LHZDK
ARC_VIRTUAL_NETWORK _NO KBS LHZGE10
#
#
# PREM velocity and density model, Q model not known, Q is average along path,
# not Q in layer. Q might be distance dependent. Kappas are low values.
# SPEC KAPPA p and kappa s 0.01 0.02
SPEC Q BELOW 1 HZ, P,S 1.0 1.0
#
# depth vp vs Qp Qal Qs Qas dens
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 3.0 5.8 3.2 500. 0.7 400. 0.7 2.6
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 15.0 6.8 3.9 500. 0.7 400. 0.7 2.9
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 24.0 8.1 4.5 500. 0.7 400. 0.7 3.4
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 80.0 8.1 4.5 500. 0.7 400. 0.7 3.4
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 171. 8.0 4.4 500. 0.7 400. 0.7 3.4
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 220. 8.6 4.6 500. 0.7 400. 0.7 3.4
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 271. 8.7 4.7 500. 0.7 400. 0.7 3.5
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 371. 8.9 4.8 500. 0.7 400. 0.7 3.5
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 400. 9.1 4.9 500. 0.7 400. 0.7 3.7
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 471. 9.5 5.1 500. 0.7 400. 0.7 3.8
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 571. 10.0 5.4 500. 0.7 400. 0.7 3.9
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 600. 10.2 5.5 500. 0.7 400. 0.7 4.0
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 670. 10.3 5.6 500. 0.7 400. 0.7 4.0
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 771. 11.1 6.2 500. 0.7 400. 0.7 4.4
SPEC MODEL h,vp,vs,qp,qap,qs,qas,d 871. 11.2 6.3 500. 0.7 400. 0.7 4.5
#
# for geometrical spreading
SPECTRAL GEO_DEPTHS 10.0 14.0
HERKIJ_DISTANCE 100.0
#
# for FOCMEC
FOCMEC MAXSOL max solutions 125.
#
# for MAP program and map option in EEV, also GMT
MAP_LAT_BORDER dist from center 3.0
MAP_LON_BORDER " 6.0
EPIMAP_STATIONS plot stations
EPIMAP_MAP_FILE name of map EUROPE
EPIMAP_PROJECTION number 3.
GMT_GRIDFILE c:\a\gmt\newbathy.grd
#
# for EEV related
EEV_COMMENT Comment for EEV Depth has benn fixed to 10 km
CHAPTER 3. INSTALLATION

EEV_COMMENT  Comment for EEV    Depth has been fixed to 20 km
REG_KEEP_AUTO Keep phases when reg  1.
MAILX COMMAND ALERT MAIL
TEXT_PRINT  Print command s-file nenscript -Psps
AUTO_PROCESS Run prog. when reg.
# Parameters for plotting type P(icture) files in EEV
PLOT_PDF_COMMAND for linux  evince
#PLOT_PDF_COMMAND for windows  start acrord32.exe
PLOT_PICTURE_COMMAND  for linux  display
#PLOT_PICTURE_COMMAND for windows  \Programfioler\irfanview\i_view32
PLOT_PS_COMMAND for windows  gswin32
#PLOT_PS_COMMAND for linux  gv
#  hypocenter and a few others
#
HIGH_ACCURACY  Set high accuracy
# order to select magnitudes as given here from top (high priority) to bottom (low priority)
MAGNITUDE_ORDER  WGCM
MAGNITUDE_ORDER  LBER
MAGNITUDE_ORDER  WBER
#  general SEISAN
#
NEW_NORDIC_FORMAT  0.0
SEISAN_LOGGING  on, 1, 0  0.0
CURSOR  0: default, 1: cross 1.
#FONT  -adobe-courier-bold-r-normal--24-240-75-75-m-150-iso8859-1
#  parameters for gmap used within eev
#GMAP_DIR  c:/seismo/WOR
GMAP_TYPE  MAP  [MAP, SATELLITE, HYBRID, TERRAIN]
#GMAP_SIZE  2.0  2.0
#  parameters for gmap
#GMAP_ICON_QUAKE  http://maps.google.com/mapfiles/kml/pal2/icon26.png
GMAP_ICON_EXPLOSION  http://maps.google.com/mapfiles/kml/shapes/star.png
GMAP_ICON_PROB_EXPL  http://maps.google.com/mapfiles/kml/shapes/open-diamond.png
GMAP_ICON_OTHER_EVENTS  http://maps.google.com/mapfiles/kml/shapes/square.png
GMAP_ICON_MSIZE  0.5
GMAP_ICON_XSIZE  0.2
GMAP_ICON_YSIZE  0.5
#GMAP_APPEND_KML  <!-- lines to be appended to the gmap.kml file : -->
GMAP_APPEND_KML  <ScreenOverlay id="LOGO">
GMAP_APPEND_KML  <name>Info and links</name>
GMAP_APPEND_KML  <description> <![CDATA[
GMAP_APPEND_KML  Data is in Nordic format. The format is <br>
3.12. SETTING GENERAL SEISAN DEFAULTS, SEISAN.DEF

GMAP_APPEND_KML described in the Seisan manual at UIB.<br>
GMAP_APPEND_KML UIB: http://www.geo.uib.no/seismo/<br>]]>
GMAP_APPEND_KML <description>
GMAP_APPEND_KML <Icon>
GMAP_APPEND_KML <href>http://seis.geus.net/geus.png</href>
GMAP_APPEND_KML </Icon>
GMAP_APPEND_KML <overlayXY x="0" y="1" xunits="fraction" yunits="fraction"/>
GMAP_APPEND_KML <screenXY x="-0.01" y="0.99" xunits="fraction" yunits="fraction"/>
GMAP_APPEND_KML <rotationXY x="0" y="0" xunits="fraction" yunits="fraction"/>
GMAP_APPEND_KML <size x="0.1" y="0.1" xunits="fraction" yunits="fraction"/>
#GMAP_APPEND_KML </ScreenOverlay>
#GMAP_APPEND_KML <!-- end of appended lines -->

# parameters for automatic gmap
GMAP_AUTO  0: no, 1:yes  1.0
GMAP_AUTO_ICON_COLOR ff0000ff
GMAP_AUTO_ICON_MSIZE 0.5
GMAP_AUTO_ICON_XSIZE 0.2
GMAP_AUTO_ICON_YSIZE 0.5
GMAP_AUTO_LOOKAT_ALTITUDE 2000000.0
GMAP_AUTO_SHOW_STAT  0: no, 1:yes  1.
GMAP_AUTO_ERROR_ELLIPSE 0: no, 1:yes  1.
GMAP_AUTO_STAT_SIZE 1.1
GMAP_AUTO_STAT_URL http://maps.google.com/mapfiles/kml/shapes/triangle.png
GMAP_AUTO_STAT_RESIDUAL_GOOD 0.5
GMAP_AUTO_STAT_RESIDUAL_BAD 1.5
GMAP_AUTO_STAT_COLOR_BAD ff0000ff
GMAP_AUTO_STAT_COLOR_OK ff00ffff
GMAP_AUTO_SHOW_OLD_LOCATION 0: no, 1:yes  1.
GMAP_AUTO_OLD_LOCATION_COLOR ffff0000
GMAP_AUTO_SHOW_PATH  0: no, 1:yes  1.
GMAP_AUTO_PATH_COLOR ff929292
GMAP_AUTO_PATH_WIDTH 2.5
GMAP_AUTO_FILE_ACTION 0: no, 1:yes  0.
GMAP_AUTO_ACTION cp gmap.cur.kml /inetpub/www/html/seismo/nnsn
# e.g. GMAP_AUTO_ACTION ncftpput -u seismo -p passwd ftp.server /home/seismo/www gmap.cur.kml
#
# EOF SEISAN.DEF

The parameters are:

ALERT EMAIL: Give email addresses (more than one are given by repeating the key word with one new address per line) to which email command in EEV will send.

AUTO_PROCESS: Set to 1. to run REG.AUTO_PROCESS.

CONFIRMATION: Level of confirmation required for example when deleting files, 0.=no confirmation
and 1.=always confirm. (at the moment only used in eev)

CONT_BASE: Waveform database to be searched (there can be several). The base is selected as a default in base selection box in MULPLT if flag is 1.0. Example:

<table>
<thead>
<tr>
<th>CONT_BASE selected by default</th>
<th>ESK</th>
<th>1.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONT_BASE not sel by default</td>
<td>EDI</td>
<td>0.</td>
</tr>
</tbody>
</table>

CONT_YEAR_MONTH_POSTION_FILE: File name of file in continuous data base: First parameter is the start position in the file name of the 4 digit year and second parameter is the start position of the 2 digit month. This parameter is also used for normal waveform files referenced in the S-file.

ARC_ARCHIVE: Path to BUD or SeiscomP archive

ARC_TYPE: Type of archive, 0: BUD, 1: SeisComP

ARC_START_TIME: Default time(sec) to start before origin time

ARC_DURATION: Duration of segment(sec)

ARC_BY_DEFAULT: 0: not used, 1: Plot whole archive even when no archive has been specified in S-file, 2: Same as 1 but only stations with readings

ARC_CHAN: BUD or SeiscomP data that MULPLT will plot, given by station code, channel name, network code and location id. Optionally, the archive channel specification lines can have a time interval of validity. This is to avoid unnecessary checking of archive files if the archive has channels from different time periods. The format is: yyyymmddhh for both start and end time, only as much info as needed has to be specified. If start time is blank, start time is very early, if end time is blank, no end time. The start end end times are written in columns 61-70 and 71-80 respectively.

CONT_BEFORE: Minutes to read into the memory before the required start time (must be at least the length of a waveform file)

CONT_AFTER: Minutes to read into the memory after the data that is plotted

CONT_YEAR_MONTH_POSTION_FILE: Position in cont file name where year yyyy and month mm starts, all files must have the same format.

COPY_WAV_DIR: Normally, waveform files are copied to WAV when registering an event. If this parameter is set, waveform files can be copied to the WAV data base specified. Max 5 characters.

CURSOR: Select cursor: 0. for pointer, 1 for cross and 2 for crosshair (Unix only, on PC only pointer is available)

EEV_COMMENT: Comments entered can then be used with EEV command ic. There can be any number of comments, max length 40 chars.

EPIMAP_MAP_FILE: The map coordinates file to use with MAP option in EEV and the MAP program.

EPIMAP_PROJECTION: The projection number used by EPIMAP with MAP option in EEV and program MAP, see EPIMAP program for choices.

EPIMAP_STATIONS: One letter indicator for plotting stations, in first column. See EPIMAP program for codes. A is all. Used with Map option in EEV as well as MAP program.

FOCMEC MAXSOL: Maximum number of solutions in FOCMEC grid search, default is 100, however, it may be required to allow for more solutions.

FONT: Select font available on X system. To check available fonts use command ‘xlsfonts’ or ‘xfontsel’
and ‘xfd -fn font’ to display a font. (Unix only)

HIGH_ACCURACY: Setting it to 1.0 enables high accuracy operation. This parameter affects programs MULPLT, HYP and UPDATE

HYPO71_OFFSET: Apply offset in degree to station and epicenter locations, required for example when not all stations are either east or west of 0 longitude.

MAGNITUDE_ORDER: A list of magnitude type and agency (e.g., LBER for ML from agency BER) can be given to specify the order of magnitudes for selection. The order is given by listing different magnitudes from top to bottom, where the magnitude at the top has highest priority and the bottom one the lowest. It is possible to leave either magnitude or agency blank in which case the blank represents a wildcard. This parameter is currently used by the program noresv only.

MAILX_COMMAND: Full path to mailx command (Unix only), which is used by email command in EEV.

MAP_LAT_BORDER, MAP_LON_BORDER: These parameters are used with command MAP and MAPG in EEV, which plot a map centered on current epicenter. The two parameters give the distance in degrees from the epicenter that the map should be plotted. If both set to ‘0.’ EEV will ask for the parameters. Parameter is the same as EPIMAP_LAT_BORDER and EPIMAP_LON_BORDER

EPIMAP_LAT_BORDER and EPIMAP_LON_BORDER: Same as MAP....

MERGE_WAVEFORM: The network code given to waveform files merged with MULPLT when running from EEV. See MULPLT and EEV. Also used in WAVETOOL and SEISEI. Max 5 characters.

SPECTRAL_GEO_DEPTHS and HERKIJ_DISTANCE. See MULPLT spectral section for explanation. Parameters used to calculate geometrical spreading.

SPEC_KAPPA: P and S kappa values, respectively, used with spec model.

SPEC_Q_BELOW_1_HZ: P and S values (X) for how Q is a function of frequency below the parameter value X. The Q-function is as follows: if X=0.0 then Q=Q0*f**qalpha. If X larger than 0.0, then Q=Q0*(1+f/X)**qalpha where f is frequency. These parameters are only active if the spec model is used, the default is X=1.0. The default is thus no frequency dependence below 1 Hz. See also MULPLT, MULPLT.DEF and AUTOMAG.

SPEC_MODEL: Gives depth, p-velocity, s-velocity, qp0, qpalpha, qs0, qsalpha and density. One line per layer in increasing depths. Format after column 40 is 8f5.0

REG_Auto_PROCESS: Name of program to run when registering event.

REG_KEEP_AUTO: If flag set to 1.0, keep automatic pics when registering event from EEV.

TEXT_PRINT: Printer command used to print an S-file from EEV.

WAVEFORM_BASE: Name of waveform data base to be searched. Normally this is a 1-5 letter data base name. The name must be written as shown above under Par1. Not needed for the default data base. The data base must have standard WAV year-month structure. It is intended to be used if for some reason the user wants to store waveform files in an alternate WAV structure instead of the the one corresponding to the S-file data base. If a waveform file is not found in the base corresponding to the S-file base, WAVEFORM_BASE's will be searched.

WAVEFORM_DIRS: The complete path to directories where the system should look for waveform files. Searched last.

OUTPUT_DIR: Output Directory for SEISAN commands results. Default “./’
SEISAN_LOGGING: Command to turn logging on or off (1. or 0.0). Default is on.
PLOT_PDF_COMMAND: Command to plot a PDF file. Command can be 40 characters.
PLOT_PICTURE_COMMAND: Command to plot a picture file like a png file. Could be display on Linux. Command can be 40 characters.
INTERNET_BROWSER: Location and place of browser
HELP_DIR: Directory of help files, usually INF

GMAP_DIR: The directory on your computer system where gmap.html is copied to when gmap is called in eev, see also section 10.5.
GMAP_TYPE: The type of background map used by Google Maps when gmap.html is opened in a browser options are: MAP, SATELLITE or HYBRID, TERRAIN.
GMAP_ICON_QUAKE: The gmap program used this parameter to defines the icon uses to illustrate an earthquake in Google Earth.
GMAP_ICON_EXPLOSION: The gmap program used this parameter to defines the icon uses to illustrate an explosion in Google Earth.
GMAP_ICON_PROB_EXPL: The gmap program used this parameter to defines the icon uses to illustrate a probable explosion in Google Earth.
GMAP_ICON_OTHER_EVENTS: The gmap program used this parameter to defines the icon uses to illustrate all other events in Google Earth.
GMAP_ICON_MSIZE: The gmap program will plot all events with a magnitude smaller than this with the size/scale of this value.
GMAP_ICON_XSIZE: The gmap program scale the epicenter icons with the formula: scale=GMAP_ICON_XSIZE * Magnitude ** GMAP_ICON_YSIZE
GMAP_ICON_YSIZE: See GMAP_ICON_XSIZE

GMAP_APPEND_KML: With this parameter you can append your own Google Earth KML code to the output file of gmap. Note, there can only be 100 GMAP_APPEND_KML lines and the KML code must be given between character no. 41 and no. 120.

GMAP_AUTO: To enable or disable the automatic generation of kml files, use 0.0 to disable or 1.0 to enable
GMAP_AUTO_ICON_EVENT: Link to icon used to display epicenter of current event
GMAP_AUTO_ICON_COLOR: color of epicenter icon e.g. ff0000ff
GMAP_AUTO_ICON_MSIZE: The magnitude is set to this values, if it is smaller or missing, e.g. 0.5
GMAP_AUTO_ICON_XSIZE: The size of the epicenter icon is given by this formula: size=GMAP_AUTO_ICON_XSIZE * Magnitude ** GMAP_AUTO_ICON_YSIZE
GMAP_AUTO_ICON_YSIZE: See GMAP_AUTO_ICON_XSIZE

GMAP_AUTO_LOOKAT_ALTITUDE: Altitude from where the epicenter is viewed, default is 2000000.0 meter.
GMAP_AUTO_SHOW_STAT: To show the used station, use 0.0 to disable or 1.0 to enable
GMAP_AUTO_ERROR_ELLIPSE: To show the error ellipse, use 0.0 to disable or 1.0 to enable
3.13. Format description in parameter files

Since most SEISAN programs are written in Fortran, the format descriptions follow the Fortran convention. The following are the main format descriptors:

1. **I**n: Integer format: E.g. I5 means an integer with 5 digits normally right justified. A completely blank field will be read as zero. Examples:

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456</td>
<td>(Position)</td>
</tr>
<tr>
<td>1100</td>
<td>read as 1100</td>
</tr>
<tr>
<td>11</td>
<td>read as 11</td>
</tr>
<tr>
<td></td>
<td>read as 0</td>
</tr>
</tbody>
</table>

2. **F**n.k: Real number format: E.g. F7.3 is a real number occupying 7 places like 111.111 and the decimal point is 3 places from the right. Any real number can occupy the 7 places like

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>(Position)</td>
</tr>
<tr>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td></td>
</tr>
</tbody>
</table>
All of these will be read as 1.1. If there is no decimal place given, it is automatically put k places from the right. Like the number 1234 read with f4.2 will be read as 12.34. nX . Spaces. Like 5x means 5 spaces.

An: Character format: Like A5 means reading 5 characters

Combining format specifications, example

1234567890123456789 position 2f5.1,1x,a4,2x,i2
11.1 12.1 text 12

Do not use tabulator instead of blanks
Chapter 4

Seisan Explorer User Guide

This chapter refers to Seisan Explorer (SE) Version 2.7.0, compiled with Qt Version 5.12.9.

The core part of the SE code is written by Øyind Natvik, Department of Earth Science, University of Bergen, Norway. The functions are mainly written by Peter Voss.

Introduction

Seisan Explorer (hereafter called SE) is a new graphical user interface for SEISAN written in C++/QT. It is supported on Windows, Linux and UNIX platforms. In its current version, it replaces the old SEISAN graphical interface for Windows, but the intention is that SE will be the main graphical user interface for SEISAN through which most of the current processing by EEV and MULPLT will take place. However, remaking MULPLT requires a large effort so the current MULPLT is used for now. In the following description, it is assumed that the user is familiar with SEISAN.

Problems
Please report bugs and questions to seisan@geo.uib.no

Installation:

Windows:
SE is installed together with Seisan.

Linux:
SE need to be compiled on Linux/UNIX platforms. Please see section 3.10.

Mac:
SE has not been tested or compiled on Mac.

Memory requirements:

When opening a database, SE will load the contents of all S-files into memory. Opening a database of 100,000 events will require 350+ megabytes of memory.

SE works with the New Nordic Format

Parameter files for SE

SE uses the same parameter files as the rest of SEISAN whenever a corresponding program is called. The parameter file SEISAN.DEF is the only one used by SE directly while MULPLT.DEF and STATIONx.HYP are read when respective options are used. SE will first look for the file in the SE work
CHAPTER 4. SEISAN EXPLORER USER GUIDE

directory, which might not be the directory from which you started SE, but the directory specified as the SE work directory, see later. If the file is not found in the the SE work directory, it will be read from DAT. The STATIONx.HYP is treated the same way as it is ingrated with SE. MULPLT is a program running outside SE so MULPLT.DEF will first be read from directory where SE is started and then from DAT if not there.

How SE works:

Starting SE:

SE can be started in two ways: Clicking on the icon in Windows or giving the command se on the prompt line (Windows and Linux). When using the command line option, arguments can be used like for EEV:

se ,, Working with a local data base
se start date Using default data base from start date and +30 days
se start date interval ———— + interval days
se start date data base interval ———— using given data base

e.g. se 20110122 TRO 10 will load data from data base TRO for the time interval January 22, 2011 and the next 10 days.

To get the options, write se help.

SE loads S-files from a SEISAN database. Only the S-files that falls within a user specified time interval are read. You may also load an index file or a local database. In this case, the currently set time interval is ignored, and time interval is adjusted automatically to fit the loaded data All information in the S-files is stored in memory for fast access. SE is dimensioned for up to one million events. In contrast to EEV, SE can read in data for several months or years, thereby giving the user the opportunity to work with large datasets without any artificial time boundaries. A subset of the data in each S-file is displayed in rows in the Event List view as demonstrated on the front page and in Figure 4.1.

When a database has been loaded, the user can select one or several events for further action, a bit like EEV. The Event List view columns can to a certain degree be customized (EventList/Select Columns). A Log view is available where most system messages are shown.

An overview of the SE application:

The application can present two views, the Event List view and the Log view. The Event List view is shown in the figure below. The program also has a title bar at the top and a status bar at the bottom. The title bar shows the currently opened database and the number of events loaded.

The left part of the status bar shows messages. Status bar messages will only be displayed for a few seconds and then disappear since they might soon be replaces by another message. The right part of the status bar shows permanent information which is the current time interval and the current operator code.

The Event List view

When a database is loaded into SE, the Event List view will be populated (see Figure 4.1), and the first event will automatically be selected. To navigate, use mouse, arrow keys or the scroll bars. Click on an event to select it. Several events can be selected as needed using standard selection procedures. Some items in the event list view may be displayed with a grey background colour. This indicates that additional information will pop up if hovering over these items with the mouse pointer.

In the Event List view you can:
• Perform actions with selected events.

• Search for an event by time: From Event List view, just start typing the desired date (type as `yyyyymmddhhmmss`), and the event nearest in time will be the selected. Press Enter when finished searching.

• Navigate to a particular row number by typing 0 followed by the row number (what in EEV was called event number) and press Enter.

• Navigate between selected events using the commands N (next), and Shift+N (previous).

• Search event(s) by time association: Use shortcut A and all events associated with each other within given time window (default 200 seconds) are selected.

Executing actions. Right-click inside the Event List view to bring up the Event List menu. The menu shows all available actions and their corresponding shortcuts. The current actions are:

• Add line: Gives a submenu for A: Add ARC lien, H: Add hypocenter line.

• Associate events (shift A): Find events occurring within a given time interval (default 200s).

• Copy to file (C): Copy highlighted events to file `se-select.out.out` (located in work directory).

• Delete (D): Delete the selected S-files.

• Duplicate (Shift+D): Duplicate an event file.

• Edit comment lines (Shift+C): Edit comment lines in S-file.
• Edit STATIONx.HYP file (Shift+S): The STATION file associated with current event (default) or any other station file can be edited.

• Edit with text editor (E): Edit the S-file.

• EEV (<): Launch EEV with current event. Some EEV commands will be disabled and it is not possible to move to another event is EEV.

• Locate (L): Locate current event with HYP. A locator window will pop up.

• Merge (M): Merge one or more events into a destination event. Note: To mark events for merging, first mark the destination event. Then mark one or more events while holding down the Ctrl key.

• Plot with mulplt (P): MULPLT is started in all default mode so no questions are asked. Results of picks are put back in S-file and SE.

• Plot with mulplt (show plot menu) (Shift-P): The mulplt plot menu will be shown so all defaults is not used.

• Reload event (ctl R). In case of changes, reload S-file.

• Register (R): Register event. Similar questions will come up as in EEV.

• Set distance indicator (Shift+R): Changes the distance indicator in SE and the S-file

• Set event indicator (ctrl E)

• Set model indicator (ctrl M)

• Show with Google Earth (Shift+G): Creates a KML file to be used with Google Earth. Google Earth has to be launched manually with the KML file. An Internet connection is required.

• Show with map (Shift M): The selected events are written out in a file called map.out in working directory and the events are plotted with SEISAN program MAP (=automatic EPIMAP). MAP can be replaced with any other mapping program you have, it runs with command map map.out.

• Show with Seismicity Viewer (V): Opens the Seismicity Viewer with the selected events.

Other actions:

• Mark/unmark event (ctrl X)

• Unmark all (alt x)

• Refresh view (F5): Refreshes the current Event List view. Does not reload the database.

• Load event file into Explorer (F3): Reads a single S-file into SE. The S-file must belong to the currently opened database, but it can be outside of the current time interval.

• Unload event from Explorer (F4): Unloads an event from SE. This action does not remove the event from the SEISAN database.

• Select all events, (Ctrl+A). Select all events in view.

• Set filter (Ctrl+F): Apply a filter to the event list.
Note that most actions are disabled while SE is busy loading a database.

Sorting data in the Event List view.

The default sorting is by date/time. However, the eventlist can also be sorted by the contents of any other column to sort by this column. Simply click on the header of any column and all events displayed are in the order of the clicked column. The order can also be reversed by clicking the header again. E.g. clicking twice on the depth column will order events by increasing depth. Columns with text will be sorted alphabetically.

Event List menu

Under this menu we find options for dealing with the Event List view. They are:

- Set filter. This means setting criteria for selection of a subset of the Event List View, see details below. E.g., all events larger than a particular magnitude can be selected and only those will then appear in the Event List view.
- Remove filter. Return to the original unfiltered view.
- Select columns. Select columns to show.
- Auto resize columns. Automatically resize the column widths (based on what is currently visible in the view).

The event selection filter

SE has an event selection filter similar to SELECT, but with many more options. Since the selection is done with data in memory, it is very fast compared to any other method of event selection. After a selection is made, the selected events are shown in the Event List View, and further processing can be done with these events. Selections are made using logical expressions built from tokens.

The selection filter has two parts, one filter to include events, and one filter to exclude events. When applying the selection filter, the include expression is executed first and will return a list of all events that matches that expression. Next, the exclude filter will be applied if it has been defined. The exclude filter will be applied only to the result of the include filter, thereby reducing the resulting event list further. This can e.g. be used to select all events between 4 and 8 in magnitude but exclude events with magnitude 5 to 6. Both filters are defined in the same way with logical expressions. See examples below.

To create a filter, press Ctrl-F, or use the Event List menu to bring up the filter dialog (see figure 4.2). Click the Edit buttons to bring up the expression builder.

The expression builder is used to define the include/exclude expressions. In expression builder, you may get explanation and examples for each token by right-clicking the token button. An include expression is mandatory. The use of an exclude expression is usually optional. When you are done creating and testing the filter, you may save the filter to a .flt file. The filter can then be loaded again later as needed. The filter’s description is also saved.

The following token types are used in expressions:

- Properties
- Operators
- Values
Figure 4.2: Filter dialog.
Figure 4.3: Expression builder.
Functions

Properties:
These are properties in the S-file, like latitude, agency and magnitudes. Property names always start
with a dollar sign, like $Agency.

Values:
Values are numbers or strings.

Operators:
The following logical operators are supported: =, <, <=, >, >=, AND, OR. In addition parentheses are
also supported.

Functions:
Functions take a set of parameters. The return value is the logical value 'true' or 'false'. All functions
take their parameters inside square brackets.

An example of a simple expression is

$L$Lat > 55 AND $L$Lat < 70 AND $L$Mag > 2

which selects events between 55 and 70 degrees N and with magnitude larger than 2.

An example of an expression with a function is

$Agency = "ber" AND InRange[$L$Lat,30,60]

which selects all events with agency 'ber' and latitude from 30 to 60.

Rules for writing a valid expression:

- The expression generally has to be constructed by one or more logical statements of type $property = value$. The statements must be bound together by AND/OR operators to form an expression that evaluates to true or false when executed.

- Spaces are needed around all tokens (except for the right/left parenthesis).

- Capitalization of the expression is generally not important, so $agency = "ber"$ is equal to $AGENCY = "BER"$.

- String values are treated in a case-insensitive way.

- String values must be enclosed in double quotes.

- Decimal numbers are not required to have a decimal point. So '10.0' can be written as '10'.

- Arguments to functions are separated by commas.

- A function may take a logical expression as an argument. These expressions must adhere to the same rules as a normal expression.

The expression builder will check the expression for errors when pressing the OK button. If any errors are detected, an error message will appear, and the error will have to be corrected.
There is a special keyword 'IncludeAll', that can be used in the include filter to include all events. If this keyword is used then an exclude expression is mandatory. Also note that this keyword must be used alone. It cannot be combined with other tokens.

Magnitudes in selection filter.

Since most events has several types of magnitudes, it can sometimes be difficult to select on magnitude criteria. SEISAN therefore optionally has a unique magnitude (or main magnitude) for each event called M. M is assigned to one of the magnitudes given for the event according to criteria for event type and event agency as set up in SEISAN.DEF. The selection criteria can use either the unique magnitude or a combination of magnitudes types and magnitude agencies.

The Log view.

The Log view (Figure 4.4) shows all important messages from SE. Information messages are black, warnings are blue, and errors are in red colour. Every time an error occurs, it will be logged here, and SE will automatically switch to the Log view to make the user aware of the error. The log view has a menu that can be activated by right-clicking anywhere in the view.

The menu has commands for common tasks such as clearing the view, or saving it to a file. If the Log view holds lines with errors/warnings for S-files, then it is possible to edit these S-files directly from the log view by right-clicking the line.

The work directory.

SE will always use a specific work directory where the log file and all output is written, e.g. from hypocenter locations. This work directory is independent from where SE is started, whether from the desktop or a particular directory. The work directory is by default WOR, but an alternative directory can be set (File/Set Work Directory). The work directory is also the local directory where MULPLT
will look for waveform files irrespective of from which directory SE has been started. The current work directory is shown in the Log view when SE is started. Multiple running copies of SE should not share the same work directory. Therefore - when SE starts up - it will look for the existence of a lock file called $\text{se-workdir}\_\text{lock}$ in its work directory. If it does not exist, then it will be created to signal that directory is now in use. If it exists, then this may indicate that another user is currently using the work directory. In this case a dialog box with a warning appears (see Figure 4.5). The user will be allowed to remove the lock file (if he is sure that none else is actually using the work directory), or select a new work directory, or exit the program. Any lock file created by SE will automatically be deleted when the program is closed. If SE should crash, then the lock file will be left in the work directory. In this case it can safely be deleted manually.

Multiple running copies of SE should not share the same work directory. Therefore - when SE starts up - it will look for the existence of a lock file called $\text{se-workdir}\_\text{lock}$ in its work directory. If it does not exist, then it will be created to signal that directory is now in use. If it exists, then this may indicate that another user is currently using the work directory. In this case a dialog box with a warning appears (see Figure 4.5).

The user will be allowed to remove the lock file (if he is sure that none else is actually using the work directory), or continue to select a new work directory. Any lock file created by SE will automatically be deleted when the program is closed. If SE should crash, then the lock file will be left in the work directory. In this case it can safely be removed manually.

Open a database.

Databases are opened under File/Open. The choices are: Default Database (as set by DEF\_BASE), Local Database, Database, Catalog File and Index File. The 'Database' option will open a dialog box showing the REA directory with all databases (standard 5 letter directories), and one can be selected. A new option - compared to EEV - is that a database (the year month structure) can be located under any directory, not just REA. It can be selected by navigating the file system from the dialog box. This arbitrary structure must have corresponding waveform files in WAV, SE working directory or a named directory (see SEISAN.DEF) and cannot use the WAV structure since it is not placed in a standard REA structure.

The 'Catalog File' option makes it possible to open a catalog file. SE will extract the S-Files inside the cat-file to a user-selectable folder, and then open the folder as a local database. When closing the database, user will be prompted to save any changes back to the catalog file. The folder will not be automatically removed when the database is closed. It will have to be deleted manually.

SE can be configured to automatically open the last used database on start-up (see File/Configure).

Local data base in SE, access to EEV and reading waveform files.
Figure 4.6: Set time interval.

Case 1: Default working directory

The default working directory in SE (example Windows) is \seismo \WOR \. A local data base can then be opened in any other directory. The waveform files must then be in WAV or WOR. If they are in the local data base directory, they can be found if parameter WAVEFORM_DIRS is set in SEISAN.DEF like WAVEFORM_DIRS Waveform directory \seismo \wor \test

where the location of the waveform files are given as \seismo \wor \test , which also can be the local data base directory.

EEV will not work since EEV will look in SE working directory.

Case 2: Directory with local data base is set as SE working directory

In SE the working directory can be set under file > File/Set Work Directory... When set to the directory of the local data base, EEV will work and waveform files will be found if in WAV or local data base directory. WAVEFORM_DIRS does not need to be set. However, no other local data base can be opened (a bug to be fixed).

Setting a time interval.

When a database is being opened, SE will suggest a default time interval. The default time interval has an End date equal to the current date, and the Start date is set to the current date minus 30 days (this value is user configurable). Both start-date and end-date can be modified by the user (see Figure 4.6). The default time span between start-date and end-date can be set under File/Configure. The time interval can be changed at any time by pressing ctrl-t or using the main menu (File/Set Time Interval). If the time interval is changed when a database is open, then the user will be prompted to reload the database.

Special functions have been written for SE, see below. To add new functions the two functions Sample
function A and Sample function B can be used. Add changed in the source code of `sample_function_A.cpp` or `sample_function_B.cpp` and recompile. If you wish to change the widget we recommend to use Qt Creator, see [http://qt-project.org/wiki/Category:Tools::QtCreator](http://qt-project.org/wiki/Category:Tools::QtCreator). Sample function A show how to get values in the event list. Sample function B show how to use the plotting widget QCustomPlot (see also [http://www.workslikeclockwork.com/index.php/components/qt-plotting-widget/](http://www.workslikeclockwork.com/index.php/components/qt-plotting-widget/)).

**Functions**

The functions are operations dealing will all data in the Event View, and not the data manually selected. The functions are:

- **With the function Gutenberg-Richter relation** a histogram showing the number of events in selected magnitude intervals can be plotted. Furthermore, the b-value can be computed from the incremental and cumulative values using linear regression. An example is seen in figure 4.7.

- **With the function Poisson distribution**. It can be tested whether or not the earthquakes in the event list are Poisson distributed. The function plot a histogram of the number of earthquakes in yearly intervals, and its corresponding Poisson distribution. An example is seen in figure 4.8.

- **With the function Completeness check** the years of completeness for selected magnitude intervals can be estimated. The output of the analysis done with this function is stored in the function `se-completeness.out`. An example is seen in figure 4.9.

- **With the function Weichert method**, the b-value can be estimated. This function use the file `se-completeness.out` as input (see 4.9). An example is seen in figure 4.10.

- **With the function Temporal-spatial hypocenter distribution**, the latitude or longitude or depth of the events in the event list can be plotted over time. An example is seen in figure 4.11.

- **With the function change-event-type** the events in the event list can be changed to E-explosion or P-possible explosion or V-volcanic or Q-known earthquake. An example is seen in figure 4.12.

- **With the function change-model-indicator** the earth model of events in the event list can be changed. An example is seen in figure 4.13.

- **With the function mag-vs-mag** magnitudes can be compared. An example is seen in figure 4.14.
Figure 4.8: se-poisson

Figure 4.9: se-completeness
CHAPTER 4. SEISAN EXPLORER USER GUIDE

Figure 4.10: se-weichert-method

Figure 4.11: Tempo-spatial hypocenter distribution.
Figure 4.12: These event types are available "E", "P", "V", "Q" and blank, here the "Q" is shown. se-change-event-type

Figure 4.13: se-change-model-indicator
Figure 4.14: se-mag-vs-mag

Figure 4.15: se-events-per-year
4.1 SE Locator

SE has got a completely new interface for locating events in SEISAN version 11.0. While previously, location was done in a separate program, the location routine (still using HYP) has been integrated into SE. This makes it possible to get a more user friendly graphical output from the locator, which can stay open while other activities take place. The locator can be started in two ways:

From the main event list, location is still started with command ‘l’, and a new window pops up.

Plotting in MULPLT, multi trace mode,’ l’ can also be used as before, but now the locator window comes up.

Figure 4.20 shows an example

With the function events-per-year the number of events per year can be plotted in selected intervals. An example is seen in figure 4.15

With the function Time of day the number of events in the event list can be plotted with respect to the hypocenter time. An example is seen in figure 4.16

With the function Time of day versus time of year one can check, if e.g. there is an increasing number of explosions at noon in a specific period. An example is seen in figure 4.17

With the function Time of day versus distance one can check if e.g. a mining area is making explosions. An example is seen in figure 4.18

With the function Simple Waveform Viewer waveform files in the Ascii Helmberger format. Use OutW command in MULPLT to generate waveform data file mulplt.wav in Ascii Helmberger format. An example is seen in figure 4.19
Figure 4.17: Time of day vs time of year
4.1. SE LOCATOR

Figure 4.18: Time of day vs distance

Figure 4.19: Simple Waveform Viewer
Figure 4.20: Example of locating an event, the locator window.
Main locator window

When the Locator window pops up, a location has already been made and the results of the location is shown. These results are the same as given when locating with EEV, however more results are shown than locating with EEV, where one has to look in the print.out file to see all results. The results are latitude, longitude, depth, origin time (all with errors). \( D_x(km), D_y(km), D_z(km) \) and \( D_{ot} \) (sec) are the difference in location from previous solution in longitude, latitude, depth and origin time. The GAP and number of stations (NSt) are also shown. The upper right corner shows the magnitude/settings window, see below.

Update and event selection

The location, as shown in the locator window, must be saved (use button Save) in order for the results to saved in the data base (S-file). It is thus possible to do test without committing the results to the data base. In the Locator window, it is also possible to go to next and previous event without returning to the main window. However, when going to next event or previous event, no location is done and the data shown is taken from the S-file. Locate must be pressed to locate.

There is also an option for Save and Rename under File tab. This option will update the S-file name and the ID line to correspond to the origin time. Outside SE this can only be done with program UPDATE, see UPDATE to see why this is done. Option ‘u’ in EEV is not changing the filename and ID, only updating result in S-file.

Under the tab Edit, it is possible to copy the epicentre location to the paste buffer for use elsewhere.

Using the locater window as an editor

Several parameters in the locater window can be edited to facilitate simple changes and tests. Many of these fields are outlined with gray like the phases and the input phase weight. In addition, the following can be edited: latitude, longitude, depth, origin time, distance indicator and event indicator. The fix depth and fix epicentre flags can be set with tick boxes. More options are available in Settings seen in the magnitude box, see Figure 4.21.

Travel times and residuals

The residuals are color coded. The limits are hardwired and the intention is that they should be user settable in the next version. The limits are:
RMS is color coded in a similar way. If any problem with the phases, the name of the Phase shows an exclamation sign (!) as shown above. This is also the case with Magnitude and Azimuth tabs.

Abbreviations are: Stat: Station code, Comp: the two letter component code from S-file, Dist: epicentral distance in km, Azm: Azimuth, O: Onset (I, E or blank), PhObs: Phase observed, WtObs: Weight observed, blank or 0 means full weight, 1 is 0.75 , 2 is 0.5, 3 is 0.25, 4 is zero weight, 9 means using S-P (all are HYPO71 style weights), T-obs: Observed travel time, PhCal: Phase calculated (the one the program calculate as the fastest), Ain: Angle of incidence at the source, WtCal: Weight used by the program (from 0.0 to 1.0), T-cal: Travel time calculated, Res: Residual=T-Obs - T-Cal and Imp: Importance of phase in the solution calculated as a percentage, importance of all phases should add up to 100

Magnitudes

The average magnitudes are shown in the Locator window, upper right and corner. All magnitudes available for the event are shown so magnitudes on additional header lines are included. The magnitudes are sorted in importance as specified in parameter file SEISAN.DEF. The magnitudes not calculated by SEISAN are shown with less contrast than the magnitudes calculated by SEISAN. The 3. magnitude on the header line, which might not be calculated by SEISAN, is considered a SEISAN magnitude. This magnitude is often put in an external reliable reference magnitude. The magnitudes not calculated by SEISAN (not including 3. Magnitude) are at ten of the list.

For each magnitude calculated by SEISAN, it is possible to show the individual station magnitudes with the magnitude residual by pressing the Magnitude tab. See Figure 4.22

The magnitude window show all the input used for calculating the magnitudes: Amplitude, period, coda and seismic moment (log value) as given in the S-file. Similar to the travel time residuals, the magnitude residuals are color coded in order to quickly find deviating magnitudes which could indicate bad readings or wrong calibration file. The limits are:

<table>
<thead>
<tr>
<th>Magnitude Limit</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0-0.19</td>
<td>Green</td>
</tr>
<tr>
<td>0.2-0.39</td>
<td>Yellow</td>
</tr>
<tr>
<td>0.4-0.79</td>
<td>Orange</td>
</tr>
<tr>
<td>&gt;0.8</td>
<td>Red</td>
</tr>
</tbody>
</table>

Additional abbreviations: MagType: Magnitude type, Amp: Amplitude (nm or nm/s), Coda: coda length(s), Moment: Log seismic moment (dyn cm) and Mag: Magnitude.

Azimuth (backazimuth)

If backazimuths are available, they can be seen in the Azimuth tab. See Figure 4.23

In this tab, only the weight observed (WtObs) values can be edited. The residuals are color coded. The limits are:

<table>
<thead>
<tr>
<th>Residual Limit</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-5</td>
<td>Green</td>
</tr>
<tr>
<td>5-10</td>
<td>Yellow</td>
</tr>
<tr>
<td>10-20</td>
<td>Orange</td>
</tr>
<tr>
<td>&gt;20</td>
<td>Red</td>
</tr>
</tbody>
</table>

Additional abbreviations: BazObs: backazimuth observed, AppVel: Apparent velocity (km/s) and Baz-Cal: Calculated backazimuth.
Figure 4.22: The magnitude window with magnitudes per channel, corresponding input data and magnitude residuals.
Figure 4.23: The Azimuth (backazimuth) tab.
Log tab

This tab will automatically come up if the location routine returns a message (like a missing station, errors in station file or if the event cannot be located). In the last case, it is often useful to check the print.out file to get an explanation for why the location was not possible.
Chapter 5

Using SEISAN

Once the system has been installed, it is ready to use. Usually all work should be done in the WOR directory or on a multi user system from your own directory. To move to WOR, type WO. Unless you have to do system work, it will not be necessary to move to any other directories. However to do so, just type the first two letters of the directory name like DA to move to the DAT directory. On a PC the Notepad editor is default (invoked with command Notepad from prompt), and on Linux it is the vi editor.

The system has two basic modes of operation. The first is to work interactively with the database. That means jumping around from event to event, plotting, interactive phase picking, locating, deleting, typing, editing or appending events (S-files). This mode is invoked with the command EEV, which uses several programs, controlled by a driver program and is intended for testing and editing of single events. Once the input data seems OK, the second mode of operation can be used.

The second mode is more like traditional data analysis where single programs are made to work on the whole or part of the database. In this mode the updated S-files and CAT-files are created. Examples are also plotting of epicenters, waveform data or searching for data fulfilling certain criteria.

The system comes with a test data set from different networks, mainly the Norwegian National Network for the time period 1990 to 2011. The data has waveform data in different formats. The data set includes events from both local and teleseismic distances. The installation of test data is separate from installation of SEISAN. However, two of the events, from June 1996, are always installed as part of the software installation.

If you want to try the system, go directly to section 5.3 to get a feeling for how the system works. You can then work directly with the data from 1996.

User guide
A short user guide is found in the following section 5.1.

Tutorial
The document 'SEISAN tutorial' is an introductory tutorial for new SEISAN users. It does not require installation any test data since only the test data included with SEISAN is used. It is given as a PDF document in INF. Or here:
http://seisan.info/seisan-tutorial.pdf

Computer exercises

67
The document ‘Introduction to SEISAN and Computer exercises in processing earthquake data’ which is a more complete tutorial for both new users as well as experienced users, is included in the distribution. The testdata used in the exercises need to be installed, see chapter 3. It covers a lot more topics than the tutorial. The document is given as PDF file (seitrain.pdf) in the INF directory. Or here: ftp://ftp.geo.uib.no/pub/seismo/SOFTWARE/SEISAN/seitrain.pdf

WEBINAR

To get an idea of how SEISAN works, new users can take a look at a SEISAN webinar that was arranged by IRIS in 2014 (see https://www.youtube.com/watch?v=KJH3ktGL_K0 or fileseisan_webinar_iris_2014.mov on the SEISAN ftp server).

Illustration of earthquake ananlysis using SEISAN

SEISAN also has a document illustrating the steps involved when analyzing seismic records. This is more a demonstration than a course. Examples are given for local, regional and teleseismic earthquakes. This is done by giving representative screen plots together with explanatory texts that follow the figures. This document is part of the New Manual of Seismological Observatory Practice manual (http://bib.telegrafenberg.de/publizieren/vertrieb/nmsop/). All the data used are also included with the test data. In SEISAN, the document is found in INF: nmsop_is11.7_seisan.pdf. Or here: http://seisan.info/INF/nmsop_is11.7_seisan.pdf

SEISAN problems

Some of the most common problems have been collected in the index under the header "Problem" and in section 5.10

5.1 Short user guide

The SEISAN manual has been divided into sections describing the individual programs. However, many tasks require the use of many programs and it is not always easy to find what can be done and which programs to use. The following section intends to give an overview of some general problems that SEISAN can work with and a list of programs to use. The following tasks have been identified:

- Routine processing: Phase picking, hypocenter location and magnitudes
- Determination of source parameters: Fault plane solution, stress drop, etc
- Determination of back azimuth and apparent velocity using arrays and networks
- Crustal structure: Velocities, layer thickness and attenuation
- Seismic catalogs: ISC data, database management, completeness, statistics, etc
- Seismic hazard: Attenuation, catalogs, and soil response

5.1.1 Routine processing

The main work of a seismic observatory is to quickly process and organize incoming data from different sources. SEISAN has a simple time ordered database (see later section) and a set of programs for these tasks. The most important programs are:
EEV and SE: The EEV and SE programs are the interactive programs for working with single events in the database. The programs are used to navigate in the database to find a given event as well as for housekeeping (splitting, merging and deleting events). Once an event has been selected, a large number of options are available like phase picking, earthquake location, fault plane solution, macroseismic information etc. All results of the interactive processing are stored in the database (S-files). EEV and SE have similar options and it is possible to use EEV options from SE. EEV is working from the prompt line while SE is using a graphical interface.

MULPLT: This is the general plotting and signal analysis program and can be used to pick phases and amplitudes, correct for instrument response, produce Wood-Anderson seismograms for determining M1, simulate WWSSN SP and LP records, determine azimuth of arrival for 3 component stations, make spectrograms, rotate seismograms, display theoretical arrival times for IASP91 phases to help identifying global phases and do spectral analysis. And more. MULPLT can be used from EEV and SE or as a stand-alone program.

FK and PFFT Determining apparent velocity and back azimuth using an array of a local/regional network.

HYP: This is the general program for hypocenter location and is based on HYPOCENTER [Lienert et al., 1986] [Lienert and Havskov, 1995]. The program can use nearly all common crustal and global phases (8 character ISC codes), locate teleseismic events using the IASP91 model and use observed azimuth and apparent velocity. The program can therefore be used with all types of input data whether from single stations or arrays. HYP can be used from EEV or as a stand-alone program. HYP is integrated into SE proving a superior interface giving more feedback and options to the user, than the prompt operation. Apparent velocity is currently only used for starting location.

EPIMAP: This is the general hypocenter plotting program for making epicenter maps and hypocenter profiles. The hypocenters can be plotted with elliptical error ellipses and EPIMAP can also be used for interactive selection of events in polygon areas. For plotting hypocenters, there is also an interface to GMT. MAP is a simple version of EPIMAP requiring no input of parameters, just the Nordic file with the hypocenters. A similar program using GMT is MAPG. Fault plane solution can also be plotted.

BUL: The function of this program is to produce a bulletin. The user can tailor the appearance to local needs and the program can produce bulletins of hypocenters only or both hypocenters and phase readings.

In addition to the above programs, several programs are available for database creation, input and output of large data sets and conversion and manipulation of waveform data.

In order to get an idea of how routine processing works, some examples of routine processing will be given below.

**Case A:** Telemetry network with 32 channel central recording

The network generates waveform event files, which are transferred to SEISAN. The tasks are:

1. Convert waveform files to SEISAN format or any of the other formats used by SEISAN. It is likely that the format is MiniSEED in which case no conversion is needed. (many events can be converted in one operation). Inspect events with MULPLT. From MULPLT, false triggers are deleted and real events are put into the database. Events are at this stage identified as local, regional or distant. Phase picks can be done at this stage, but is usually done later.

2. Interactive phase picking, earthquake location, magnitude etc done with EEV or SE. Automatic phase picking is also possible at this stage.

3. Database is updated (UPDATE for many events, outside EEV or SE or one by one in EEV or
SE) once a suitable interval has been processed interactively, usually a month. Updating means permanently storing the updated or newly calculated hypocenters etc in the database.

4: Make hypocenter maps with EPIMAP, MAP or MAPG.

5: Produce a bulletin with BUL.

**Case B:** 3 telemetry networks and one broad band station

The routine is the same as above except for one additional step between 1 and 2. Since several data sets are available, some of the detections from different networks or the broad band station might correspond to the same event. There are now two options. The first is to merge the waveform files for corresponding events and then put the events into the database. The second option is to put all real events into the database and then do the merging from EEV or SE.

**Case C:** A mix of stations and networks and additional phase readings

The steps are as in case B except that before step 2, the additional phase data is put into the database. In this case the merging of events must be done with EEV.

**Case D:** A network recording all data in continuous mode into a SEISAN continuous data base or now more likely a SeisComP continuous data base. In addition, there is likely to be network wide triggering put into SEISAN. In this case it is a question of inspecting the triggers with EEV or SE/MULPLT as above and possibly extract additional data out of the continuous data base and put it into the data base. another option, which now is becoming more common is to put a reference into the S-file where to find the data in the continuous data SeisComP database so it is possible to plot the corresponding waveform data without having any event waveform files.

It should be noted that data collection and step 1 to 3 is fully automated using SEISNET [Ottemöller and Havskov, 1999]. However, with new continuous networks, SISNET is no longer the best option.

Example of using EEV for interactive processing:

Find event in default database nearest the given date and time: EEV 1999020303

Once EEV is started, an EEV prompt is given and different EEV options are available. Examples are:

E: Edit event, P: Plot event, L: Locate event, F: Make fault plane solution, d2201: Find event nearest day 22 at 01 hour, MAP: Start EPIMAP to show earthquake location and SAC: Start SAC processing of event using all parameter and waveform data from SEISAN database.

The above examples have mostly described the interactive processing of single events. However, once the data is in the database, operations can be done on the whole database, for any time interval or for events fulfilling certain criteria (like magnitude, area etc). Examples are relocating events, extracting data and determining coda Q.

5.1.2 Source parameters

The routine processing normally produces magnitudes and hypocenters. The fault plane solution can be determined using polarities and one event (Snoke et. al., 1984). Composite fault plane solutions can also be made. In addition there are 3 more programs for making fault plane solutions, 2 of which also can use amplitude ratios. A second way of determining fault plane solution is to synthetically model the waveforms using the modeling programs. In addition, seismic moment, stress drop and seismic source radius can be determined by doing spectral analysis or spectral modeling. This can also be done automatically with AUTOSIG, AUTO and AUTOMAG.
The full wave modeling programs integrated with SEISAN, are written by Bouchon [1981] and Herrmann (Herrmann,1996). The ray-tracing program is based on WKBJ and written by Chapman et al. [1988] and integrated with SEISAN by Valerie Maupin. All the above programs are executed from EEV in order to use known source parameters.

Momnet tensor inversion can be made in EEV using the Dreger (2003) programs.

5.1.3 Crustal structure and Q

A large database can be a good source of information for determining structural parameters and SEISAN provides several programs to determine the crustal structure and Q. Using seismic arrival times, it is possible to invert for the crustal structure using the VELEST program [Kissling et al., 1994]. It is also possible to do forward modeling using the location program for a large number earthquakes, since it at the end of a run, a summary of average station travel time residuals and event RMS is given. A special option of HYP is to locate a data set with all permutation of a given range of models in order to find the model giving the lowest RMS.

Deep earthquakes under a local network produce clear phase conversion at crustal interfaces [Chiu et al., 1986]. They can be modeled with one of the full wave modeling programs both with respect to amplitude and arrival time.

SEISAN can, when displaying surface waves, make spectral files ready to be processed for surface wave dispersion with Herrmann’s programs (Herrmann, 1996). This ha not been tested for a long time!

Attenuation can be determined using the coda Q method for local earthquakes (CODAQ). The coda Q program will calculate q for a series of events and stations at given frequencies. Average values are calculated and a q vs f curve is fitted to the calculated values. The principle for calculation is the standard coda q method, whereby a coda window is bandpass filtered, an envelope fitted and the coda q at the corresponding frequency calculated [Havskov et al., 1989]. The SPEC program will determine Q by calculating spectral ratios or the near surface attenuation using the spectral decay method. An alternative is to use spectral modeling where Q, stress drop and seismic moment are modeled simultaneously. The AUTOMAG program can do a grid search for find the best attenuation parameters that will fit a series of earthquakes.

The QLG program will make an inverison of the Lg spectra for a series of earthquakes to find Q.

5.1.4 Magnitudes in SEISAN

Magnitudes are dealt with in many different programs in SEISAN and this section intends to give an overview.

SEISAN can calculate most standard international magnitudes: Local magnitude Ml, coda magnitude Mc, body wave magnitudes mb and mB (broadband), surface wave magnitudes Ms and MS (broad band) and moment magnitude Mw. The magnitudes are calculated in HYP as median of the station magnitudes. Optionally, also averages can be calculated. For more details on the parameters used and the exact formulas, see program HYP.

Parameters for magnitudes

The input parameters used for calculating magnitudes are amplitudes and periods, code length and seismic moment M0. Amplitude and coda length can be obtained manually from MULPLT while automatic amplitudes for Ml can be obtained with programs EEV (command am running AUTOMAG), programs
AUTOSIG and AUTOMAG. Seismic moments are obtained from spectra manually with MULPLT or automatically with AUTOSIG and AUTOMAG. Moments can also be obtained from moment tensor inversion using EEV. The amplitudes are stored as phases with the following names: Ml: IAMl mb IAMb mB IVmB BB Ms IAMs 20 MS IVMs BB

while the moments obtained by spectral analysis are stored in the SPEC lines and in the MOM lines for moment obtained by moment tensor inversion. While the magnitudes are calculated for all channels, only the average magnitude is stored on the header lines (except for Mw from spectral analysis, where magnitudes corresponding to each moment is stored on the SPEC line). The parameters to calculated Ml and Mc are given in STATION0.HYP file. The parameters for calculating spectral magnitude are given in MULPLT.DEF and/or SEISAN.DEF. Parameters for calculating Mw from moment tensor inversion is generated by EEV (using the model in STATION0.HYP) and stored in the S-file. The remaining magnitudes have fixed parameters.

Changing Ml and Mc parameters will result in new magnitudes being calculated with the next update of the S-file. For the spectral magnitudes, only changes caused by hypocentral distance, density and velocity will affect the recalculation of magnitude. If the attenuation change, the spectra must be recalculated.

Storing magnitudes

Magnitudes are written on the type one header line. On the main first header line there is room for 3 magnitudes, see example below

```
1998 812 1410 26.8 L 36.755-121.462 8.0FFBG 4 1.3 4.6LTES 4.9WTES 5.0WHRV
```

where magnitude is given by type (e.g. L for Ml), size and a 3 letter agency. When the location program HYP is operated, the magnitudes are overwritten EXCEPT for the 3. magnitude on the header line which never will be deleted by any program. The intention with this is to be able to store an external reference magnitude. If more than 3 magnitudes are calculated (2 if a magnitude in 3. position), a new type one line is written with exactly the same origin time and location agency

```
1998 812 1410 27.0 L 36.755-121.462 8.0FFBG 4 1.7 4.6LTES 4.7CTES 5.0WHRV
1998 812 1410 27.0 L BGS 4.9WTES 1
```

There is thus room for only 6 magnitudes calculated by SEISAN. More magnitudes can be stored on more type 1 lines which must have different location agency and/or origin time compared to the first header line.

Searching for events using magnitude criteria

Program SELECT can search for a combination of magnitudes of different type, size and agency. SELECT can also optionally search for magnitudes on all header lines. SeisanExplorer (SE) can also search for a magnitude combination, but only among the 6 prime magnitudes. Manipulating magnitudes

There can be a need to convert one magnitude to another. The MAG program can make magnitude relations (also using all header lines). This magnitude relation can then be used, also by MAG, to convert one magnitude to another and write it back to the S-file. Since some programs (like EPIMAP) uses the magnitude in the first position on the header line, MAG can also be used to move any one of the magnitudes to that position. Program NORHEAD can move magnitudes from following header lines up to empty magnitude spaces on the first line. Program REPORT can move magnitudes around on the header line according to user choices. This moving around of magnitudes is now not needed for some programs. There is now a magnitude in SEISAN simply called M, which is unique magnitude returned according to priorities given in SEISAN.DEF
5.1. SHORT USER GUIDE

Magnitude order

where the order to select magnitudes as given here from top (high priority) to bottom (low priority). Only a few programs use this facility: SE, CLUSTER and ASSO. Plotting magnitude information

Magnitude sizes are plotted on all epicentral and hypocentral plotting programs. Relations between magnitudes can be plotted with MAG, b-value can be calculated and plotted with SE (standard method and Wiechert method) and BVALUE. A completeness check can be plotted with SE.

Other programs using magnitudes

NORHEAD can take magnitudes from following header lines and move them up to empty magnitude spaces on first line.

CLUSTER cleans a catalog for foreshocks and aftershocks, partly based on magnitudes. ASSO is a program that merges events in two catalogs based on both time and magnitudes.

Calculating magnitude relations

The Ml attenuation function can be calculated by inverting amplitude reading from many events suing program MAG2. A coda wave relation can be made with MAG. Attenuation to be used with spectral Mw can be determined using QLG, SPEC and CODAQ. QLG is the recommended program. AUTOMAG can also be made to make some initial tests for the best Q to fit the Brune spectrum.

The magnitude implementation in SEISAN has been adjusted (version 8.3) to the new IASPEI standard.

Amplitude based magnitude identification

Over the years there has been different ways of identifying the amplitudes used for magnitudes. Originally, amplitudes were not identified with any specific magnitude and only the period and distance was used to find out which kind of magnitude should be calculated, like Ml or Ms. Later different amplitude type phase names were used to identify which type of magnitude should be calculated and now only one IASPEI name is used for the amplitude phase name. However, SEISAN is backwards compatible and will use all the old amplitude phase identifiers. Below is given all identifiers used.

Local magnitude ML:

- Period range: $T < 5\,\text{s}$
- Distance range: distance $< \text{TEST}(57)$
- No depth range, distance is the hypocentral distance
- Phases accepted: Blank, L, S, Sg, SG, AMPL, AML, AMP and IAML (standard)

Body wave magnitude mb:

- Period range: $0.2\,\text{s} < T < 3\,\text{s}$
- Distance range: TEST(113) $< \text{distance} < 100^\circ$, before version 10.5 the lower distance was TEST(57)
- Depth range is handled by the attenuation function, SEISAN has two attenuation functions TEST(108) is used to specify which one is used. Default TEST(108) is 0 the Gutenberg and Richter [1956] attenuation function as recommended by IASPEI.
CHAPTER 5. USING SEISAN

- Phases accepted: Blank, P, AMPB, AMB, AMPb, AMb, AMP, IAmb (standard)

Surface wave magnitude Ms (Ms20):

- Period range: T > 10 s
- Distance range: TEST(114) ≤ distance ≤ 160°, before version 10.5 there was no distance limits. TEST(114) is 20° by default
- Depth < TEST(115), before version 10.5 there was no depth limit. TEST(115) is 60 km by default
- Phases accepted: Blank, AMPS, AMS, AMP, IAMs20 (standard)

Broad band body wave magnitude mB (mB_BB):

- Period range: 0.2 s < T < 30 s
- Distance range: TEST(113) < distance < 100°, before version 10.5 the lower limit of distance was TEST(57). TEST(113) is 20° by default
- Depth: All depths values are accepted.
- Phases accepted: IVmB_BB

Broad band surface magnitude MS (Ms_BB):

- Period range: 3 s < T < 60 s
- Distance range: 2° ≤ distance ≤ 160°
- Depth < TEST(115), before version 10.5 there was no depth limit. TEST(115) is 60 km by default
- Phases accepted: IVMs_BB (standard)

Several parameters used for magnitude determination is controlled by the TEST values set in the STATION0.HYP (local events), STATIOND.HYP (distant event) or other model dependent STATION file. See the HYP chapter 7.1 for default TEST values. The above values are:

- TEST(57) 1500km
- TEST(113) 20°
- TEST(114) 20°
- TEST(115) 60 km

The underlined values are hardcoded in SEISAN.

Example showing mb, mB, Ms and MS amplitude and period readings with estimated magnitudes, on the BSD station respectively:

In old Nordic:

2015 1117 0710 30.8 D 39.919 20.421136.3 DNK 21 0.6 6.5sDNK 6.1SDNK 5.8bDNK1
STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO AIN AR TRES w DIS CAZ7
BSD HZ IAm 714 18.53 194.3 0.98 1739 348
BSD HZ IVMs_BB 714 14.54 2995.7 3.96 1739 348
5.1. SHORT USER GUIDE

In new Nordic:

<table>
<thead>
<tr>
<th>Year</th>
<th>ID</th>
<th>Type</th>
<th>Distance</th>
<th>Magnitude</th>
<th>Depth</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>2015</td>
<td>1117</td>
<td>710</td>
<td>30.8 D</td>
<td>39.919</td>
<td>20.421</td>
<td>136.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Hyp output:

<table>
<thead>
<tr>
<th>Site</th>
<th>Distance</th>
<th>Magnitude</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSD HZ</td>
<td>1740.0</td>
<td>194.3</td>
<td>T: 1.0 mb = 5.4</td>
</tr>
<tr>
<td>BSD HZ</td>
<td>1740.0</td>
<td>2995.7</td>
<td>T: 4.0 mb = 5.7</td>
</tr>
<tr>
<td>BSD HZ</td>
<td>1740.0</td>
<td>97917.0</td>
<td>T: 18.0 Ms = 6.0</td>
</tr>
<tr>
<td>BSD HZ</td>
<td>1740.0</td>
<td>43913.8</td>
<td>T: 12.3 Ms = 6.1</td>
</tr>
</tbody>
</table>

5.1.5 Catalog and database work

Once a large database has been created, several programs are used to manipulate and analyze the data. The catalog can be searched for a large number of parameters. Selection criteria are: Magnitude range, magnitude types, event types (e.g. local, distant, volcanic, explosion), latitude, longitude and depth range, RMS of travel time residuals, number of stations used in the location, felt events, number of polarities, presence of certain stations etc. Events can also be selected in an area with the program used for hypocentral plots. SE has an even more sophisticated search facility.

A very useful source of data is the ISC. Data from ISC CD ROM’s can be read and converted to SEISAN format (hypocenters and phase data) and put into a database. However CD’s are little used now and a better way is to download the data in ISF format (soon ISF2 format) and convert to SEISAN with IMSNOR. The data can then be used for e.g. seismic hazard, fault plane solution or it can be relocated. A general task with catalogs is to homogenize magnitudes. Magnitude relations between e.g. Mb and Ms or Ms from one agency to Ms from another agency can be done with the program MAG. The program will also convert one magnitude to another once the linear regression has been determined. Hypocenters from different agencies can be compared with COMPARE_HYP. This program can also be used to compare hypocenters and magnitudes from two files containing the same events but eg located with different models. Event statistics can be made with STATIS and b-values calculated with BVALUE (good function for that also in SE). The number of events as a function of time is plotted with CATSTAT.

5.1.6 Seismic hazard

Probabilistic earthquake hazard computations is done, using the EQRISK program [McGuire 1976] or the CRISIS2012 program [Ordaz 1991, 1999]. EQRISK computes seismic hazard in terms of probabilities of exceedence vs earthquake intensity measures such as peak ground acceleration (PGA), for a given site or a grid of sites for up to eight different return periods. The site amplification is calculated with the
SPEC program. This is used for making spectra of many seismic signals in a semiautomatic manner. The program is intended for two purposes: (1) making relative spectra for a series of pairs of stations terminated by the average spectra, (2) Making a series of spectra for a number of stations and events. The spectra can be corrected for distance, q, and instrument response. This section involves a large number of programs and a more detailed description is given in section 34.

5.2 Getting data into the database

The first requirement for interactive work with the event editor EEV or SE is to get the data into the database.

There are two ways to get data into the database, as described in section 5.2.1 and 5.2.3. It is of course possible to make the individual S-files directly in the REA directories with the editor. This would be rather slow, and be against the philosophy of the system. However, it is mentioned in order to point out how simple the database structure is.

The SEISAN system can be used with or without digital data, the only difference to the directory structure is that the WAV and CAL directories are present when using digital waveform data. However the way of getting data into the database differs in the two cases and will be described separately.

5.2.1 System with digital data

This means that the original data is individual digital event waveform files generated by some data acquisition system. The waveform data can be stored in SEISAN, MiniSeed, GSE and SAC format as single or multi trace files. The files that are used in conjunction with the database are normally stored in WAV but can also be in the user’s directory, e.g. WOR. The normal scenario would be that multiplexed files would be transferred from a digital field station, demultiplexed and converted to SEISAN or other waveform format, preferably MiniSeed. Programs are provided to convert from most of the popular waveform formats like MINISEED, GSE, PCSUDS and from commercial recorders. It is most practical to initially put the files in WOR, check the events for false triggers, save the true events in WAV, make the corresponding S-file and a hardcopy of the digital data. All of this can be done with the program MULPLT. The program plots channels from a single waveform file. The user can then interactively decide if this is an event to keep, in which case an S-file is created in the database and the event is moved to WAV.

Alternatively, all new waveform files can be auto-registered into the database (AUTOREG) and all checking takes place from EEV or SE.

When digital data is the input to the analysis system, MULPLT is the program to use to get data into the database. From there on further analysis can be done with EEV (picking phases, locating and editing). MULPLT is also the program used with EEV. For more details on MULPLT, see detailed description in section 8.

Metadata: SEED or dataless SEED

See also chapter 2.2.3

To get SEED or dataless SEED metadata into a SEISAN database, you need the RDSEED program from IRIS. SEISAN include an old version of RDSEED, find the newest at: http://www.iris.edu/dms/nodes/dmc/software/downloads/rdseed/

First you need to extract the Staion information from the SEED or dataless SEED file, use the command:
rdseed -S -f <file>

Then you need to extract the response information from the SEED or dataless SEED file, use the command:
rdseed -R -f <file>

Now you must list the response file, use the command:
dirf RESP.*

Then you run the program RDSEED2SEISAN, this program give a output file rdseed.STATION0.HYP with station coordinates that you can add to your STATION0.HYP file in the DAT folder. And it gives a output file rdseed.SEISAN.DEF with ARC_CHAN lines that can be added to your SEISAN.DEF file in DAT, if you wish to store the data in an archive (e.g. BUD or SCP).

Finally you must move the RESP file to the CAL folder.

Waveform data: SEED or miniseed data
To get SEED or miniseed waveform data into a SEISAN database, you need the RDSEED program from IRIS (see URL above). And the DATASELECT program from IRIS if you wish to use an archive for the data:
https://seiscode.iris.washington.edu/

First you need to extract the miniseed data from the SEED, use the command:
rdseed -o 4 -f <file>

Then you need to send the data to the archive defined in the SEISAN.DEF file. This is done using the program DATASELECT that you need to download from: https://seiscode.iris.washington.edu/projects/dataselect/wiki
An older version of DATASELECT is found in the SUB folder.
For a SeiscomP archive, use the command for waveform data:
dataselect -A archive-path/%Y/%n/%s/%c.D/%n.%s.%l.%c.D.%Y.%j mini.seed
For a BUD archive, use the command:
dataselect -BUD archive-path mini.seed
Where archive-path must be the same path as ARC_ARCHIVE that is defined in SEISAN.DAT.

To plot the archived waveform data use the command textttarc in eev, before you call MULPLT

5.2.2 Instrument response
The response information gives the gain of a channel in counts/m and to get the correct ground displacement, the count values must be divided by the response values.
The response files can be located in CAL, or, if many files are available optionally also in a subdirectory structure.
The response files can be generated with RESP. Response files can also be imported in SEED format and SAC format can also be used.
CHAPTER 5. USING SEISAN

For details, see section 41

5.2.3 System without digital data

In this case the user would get phase data from other sources, e.g. analog seismograms or files with readings from other stations and agencies. These files are assumed to be written in Nordic Format. Conversion can be done from other formats like ISC, NEIC and HYPO71.

If a user already has a file with one or several events in Nordic Format, this file can be split up into single files which are copied (from any directory) into the database by using the command SPLIT. Creating a new file in Nordic Format can also be done with the program NEWEVE (use command NEWEVE).

The SPLIT program then reads the NEWEVE output file and writes out single S-files with correct names either in the current directory (default) or in the database specified (BER or another). The reason that the database specifically must be given is that the user should not accidentally put data into the database (see section 13).

5.2.4 Getting data from ISC into SEISAN

ISC provides access to its data via the web page www.isc.ac.uk. SEISAN has facilities for converting ISC output from either the bulletin data base which contains all data (readings etc) or just the catalog which just has the main information corresponding to the SEISAN header line.

Search complete ISC bulletin

Go to Search the ISC Bulletin/Bulletin search Select criteria, like area, time etc. Note that if phase output is chosen, too much data could result and not all is output, so do not choose too much data. You now see output on the screen like

Search summary:
Database: ISC Bulletin
Search type: Global search
Start date: 2014-03-31 00:00:00
End date: 2014-03-31 02:00:00
Events found: 98

DATA_TYPE BULLETIN IMS1.0:short

ISC Bulletin
Event 604511947 Morocco

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Err</th>
<th>RMS</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Smaj</th>
<th>Smin</th>
<th>Az</th>
<th>Depth</th>
<th>Err</th>
<th>Ndef</th>
<th>Nsta</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014/03/31</td>
<td>00:01:58.20</td>
<td>2.96</td>
<td>0.710</td>
<td>33.3883</td>
<td>-5.0333</td>
<td>27.8</td>
<td>24.0</td>
<td>158</td>
<td>0.0f</td>
<td>23</td>
<td>25</td>
<td>4.52</td>
<td>6.73</td>
</tr>
<tr>
<td>2014/03/31</td>
<td>00:01:58.22</td>
<td>0.82</td>
<td>1.030</td>
<td>33.3370</td>
<td>-4.9830</td>
<td>5.7</td>
<td>4.0</td>
<td>73</td>
<td>1.2</td>
<td>5.5</td>
<td>25</td>
<td>0.31</td>
<td>4.05</td>
</tr>
<tr>
<td>2014/03/31</td>
<td>00:01:59.20</td>
<td>1.07</td>
<td>0.400</td>
<td>33.3270</td>
<td>-4.8900</td>
<td>5.0</td>
<td>2.4</td>
<td>73</td>
<td>31.0f</td>
<td>25</td>
<td>17</td>
<td>152</td>
<td></td>
</tr>
<tr>
<td>2014/03/31</td>
<td>00:01:56.55</td>
<td>1.17</td>
<td>1.752</td>
<td>33.4408</td>
<td>-4.8850</td>
<td>5.539</td>
<td>3.543</td>
<td>86</td>
<td>8.1</td>
<td>10.2</td>
<td>68</td>
<td>41</td>
<td>80</td>
</tr>
</tbody>
</table>

Magnitude Err Nsta Author OrigID
mb    3.3 0.3 5 MDD 07221203
ML    2.0 INMG 07894096
5.2. GETTING DATA INTO THE DATABASE

Sta Dist EvAz Phase Time TRes Azim AzRes Slow SRes Def SNR Amp
CZD 0.43 197.8 Pg 00:02:05.057 0.1 T__ __ 46085918
CZD 0.43 197.8 Sg 00:02:09.642 -1.0 T__ __ 46085919
MD31 0.58 169.5 Pg 00:02:08.037 0.2 T__ __ 46085920
MD31 0.58 169.5 Sg 00:02:14.202 -1.2 T__ __ 46085921
MDT 0.67 159.5 Pg 00:02:09.692 0.3 T__ __ 46085922
MDT 0.67 159.5 Sg 00:02:16.808 -1.3 T__ __ 46085923
RSA 1.64 331.9 Pb 00:02:27.239 0.2 T__ __ 46085923

Save the screen output in a text file by copying and pasting using e.g. notepad or vi. In this example we put it into file isc.txt but it could be any name. With some browsers, there might be an option to copy directly into a text file. The default format on screen is the IMS1.0 short format which is the same as the IASPEI seismic format. There will soon be a more extensive ISF format called isf2 which also contains components ++. This can be used by program IMSNR. SEISAN cannot read QuakeML format directly but QuakeMk can be converted to Nordif format (old) with QML2NOR.

Convert with program IMSNR:

The previous program used is NORIMS, still included, but it does not make as complete conversion as IMSNR and it does not convert isf2. Details of IMSNR is found in the section on conversion programs.

Search for events only

An event catalog is available under Bulletin search/Event catalog. The criteria is much the same as above but the output format is different. SEISAN only supports CSV format. The following shows an example output.

Search summary:
Database: ISC Bulletin
Search type: Global search
Start date: 2014-03-31 00:00:00
End date: 2014-03-31 01:00:00
Events found: 37

DATA_TYPE EVENT_CATALOGUE
ISC Bulletin
--EVENT--|-------------------ORIGIN (PRIME HYPOCENTRE)-------------------|------MAGNITUDES-----...

EVENTID,AUTHOR ,DATE ,TIME ,LAT ,LON ,DEPTH,DEPFIX,AUTHOR ,TYPE ,MAG
604511947,ISC ,2014-03-31,00:01:56.55, 33.4408, -4.8850, 8.1, ,MDD ,mb , 3.3
606581842,JMA ,2014-03-31,00:02:31.30, 37.0400, 140.5100, 8.0, ,JMA ,M , 1.6
606581843,JMA ,2014-03-31,00:03:45.10, 34.0400, 135.3200, 7.0, ,JMA ,M , 0.5
608276337,ROM ,2014-03-31,00:04:01.93, 42.7815, 12.5628, 10.2, ,ROM ,ML , 1.2
607475289,ATH ,2014-03-31,00:08:47.54, 38.4053, 21.9940, 7.8, ,ATH ,ML , 1.4
608873452,ISK ,2014-03-31,00:09:47.90, 37.0850, 36.7560, 5.4, ,ISK ,ML , 2.2
607122021,TRN ,2014-03-31,00:12:09.08, 17.4540, -61.9990, 24.4, ,TRN ,MD , 2.7
608276338,ROM ,2014-03-31,00:12:55.66, 38.0828, 15.0768, 10.8, ,ROM ,ML , 1.4
606581844,JMA ,2014-03-31,00:21:07.70, 35.5700, 139.3500, 114.0, ,JMA ,M , 2.1
606442363,NNC ,2014-03-31,00:23:17.31, 43.1481, 78.3658, 0.0, ,NNC ,mpv , 2.5
608276339,ROM ,2014-03-31,00:23:35.47, 43.5047, 12.3582, 8.9, ,ROM ,ML , 1.5
### CHAPTER 5. USING SEISAN

<table>
<thead>
<tr>
<th>Event ID</th>
<th>Network</th>
<th>Date/Time</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Magnitude</th>
<th>Depth</th>
<th>Source</th>
<th>P-Phase</th>
<th>S-Phase</th>
<th>Velocity</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>606581845</td>
<td>JMA</td>
<td>2014-03-31 00:24:13.50</td>
<td>35.6600</td>
<td>140.9700</td>
<td>18.0</td>
<td></td>
<td>JMA</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>607764104</td>
<td>RSNC</td>
<td>2014-03-31 00:25:32.90</td>
<td>6.8670</td>
<td>-73.3360</td>
<td>122.0</td>
<td></td>
<td>RSNC</td>
<td>ML</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>607457290</td>
<td>ATH</td>
<td>2014-03-31 00:26:19.97</td>
<td>38.1553</td>
<td>20.3230</td>
<td>6.2</td>
<td></td>
<td>ATH</td>
<td>M</td>
<td>1.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>606581846</td>
<td>JMA</td>
<td>2014-03-31 00:26:37.90</td>
<td>35.6500</td>
<td>140.9600</td>
<td>17.0</td>
<td></td>
<td>JMA</td>
<td>M</td>
<td>2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>608276340</td>
<td>ROM</td>
<td>2014-03-31 00:26:58.27</td>
<td>44.2772</td>
<td>10.7885</td>
<td>7.8</td>
<td></td>
<td>ROM</td>
<td>ML</td>
<td>1.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>609917823</td>
<td>WEL</td>
<td>2014-03-31 00:27:07.25</td>
<td>48.1538</td>
<td>-71.3242</td>
<td>18.0</td>
<td>TRUE</td>
<td>WEL</td>
<td>Mn</td>
<td>2.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>606581847</td>
<td>JMA</td>
<td>2014-03-31 00:26:37.90</td>
<td>35.6600</td>
<td>140.9700</td>
<td>18.0</td>
<td></td>
<td>JMA</td>
<td>M</td>
<td>3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>606581848</td>
<td>JMA</td>
<td>2014-03-31 00:26:58.27</td>
<td>44.2772</td>
<td>10.7885</td>
<td>7.8</td>
<td></td>
<td>JMA</td>
<td>M</td>
<td>2.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>609917824</td>
<td>WEL</td>
<td>2014-03-31 00:27:07.25</td>
<td>48.1538</td>
<td>-71.3242</td>
<td>18.0</td>
<td>TRUE</td>
<td>WEL</td>
<td>M</td>
<td>1.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>608276341</td>
<td>ROM</td>
<td>2014-03-31 00:27:07.25</td>
<td>48.1538</td>
<td>-71.3242</td>
<td>18.0</td>
<td>TRUE</td>
<td>ROM</td>
<td>ML</td>
<td>2.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

International Seismological Centre  
Pipers Lane, Thatcham, Berkshire  
United Kingdom RG19 4NS  
+44 (0)1635 861022, voice; +44 (0)1635 872351, fax  
admin isc.ac.uk, e-mail

Save the output in a text file like above, here we use isc.csv, but it can be any name.  
Convert to SEISAN with program ISCCSV2NOR

C:\seismo\WOR>isccsv2nor  
Input file  
isc.csv  
Number of events 37  
Output file name is csvnor.out

The output file has this content

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Magnitude</th>
<th>Depth</th>
<th>Source</th>
<th>P-Phase</th>
<th>S-Phase</th>
<th>Velocity</th>
<th>Location</th>
</tr>
</thead>
</table>
| 2014 0331 0001 56.5 L | 33.440 | -4.885 | 8.1 | MDD | 3.3bINM | 2.0L | 1
| 2014 0331 0002 31.3 L | 37.040 | 140.510 | 8.0 | JMA | 1.6 | 1

---

STOP
This file is a compact file, however program SPLIT can also work on compact files if there is a need to put the data into a data base.

### 5.2.5 Getting data from USGS into SEISAN, get_web_locations

Written by Terje Utheim, terje@geo.uib.no The purpose of this program is to download hypocenter information from international data centers. The program has now been implemented for USGS but will also be implemented for ISC and EMSC.

The program downloads the data to a SEISAN data base in a Nordic format file in old format. The program downloads a file from the USGS called get_web_locations.txt from which it picks up relevant information for the S-file. The download might have data from several other sources which are entered into the S-file. The list of sources can be found at https://earthquake.usgs.gov/data/comcat/contributor/.

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>_longitude</th>
<th>_latitude</th>
<th>_depth</th>
<th>_mag</th>
<th>_nst</th>
<th>_mag</th>
<th>_nst</th>
<th>_nst</th>
<th>_nst</th>
<th>_nst</th>
<th>_nst</th>
</tr>
</thead>
</table>
| 2014 0331 0003 | 45.1 | 34.040 | 135.320 | 7.0 | JMA | 0.5 | 1
| 2014 0331 0004 | 01.9 | 42.781 | 12.562 | 10.2 | ROM | 1.2L | 1
| 2014 0331 0008 | 47.5 | 38.405 | 21.994 | 7.8 | ATH | 1.4L | 1
| 2014 0331 0009 | 47.9 | 37.085 | 36.756 | 5.4 | ISK | 2.2L | 1
| 2014 0331 0012 | 09.0 | 17.454 | -61.999 | 24.4 | TRN | 2.7D | 1
| 2014 0331 0012 | 55.6 | 38.082 | 15.076 | 10.8 | ROM | 1.4L | 1
| 2014 0331 0021 | 07.7 | 35.570 | 139.350 | 114.0 | JMA | 2.1 | 1
| 2014 0331 0023 | 17.3 | 43.148 | 78.365 | 0.0 | NNC | 2.5p | 1
| 2014 0331 0023 | 35.4 | 43.504 | 12.358 | 8.9 | ROM | 1.5L | 1
| 2014 0331 0024 | 13.5 | 35.660 | 140.970 | 18.0 | JMA | 3.4 | 1
| 2014 0331 0025 | 32.9 | 6.867 | -73.336 | 24.4 | TRN | 2.7D | 1
| 2014 0331 0026 | 19.9 | 38.155 | 20.323 | 6.2 | ATH | 1.7L | 1
| 2014 0331 0026 | 37.9 | 35.650 | 140.960 | 17.0 | JMA | 2.1 | 1
| 2014 0331 0026 | 58.2 | 44.277 | 10.788 | 7.8 | ROM | 1.6LGEN | 1.6L | 1
| 2014 0331 0027 | 07.2 | -39.009 | 176.186 | 63.9 | WEL | 2.6WEL | 2.8WEL | 2.6L | 1
| 2014 0331 0029 | 40.1 | 25.088 | 121.775 | 134.4 | TAP | 2.4L | 1
| 2014 0331 0030 | 29.7 | 38.327 | 20.430 | 10.2 | ATH | 1.4L | 1
| 2014 0331 0030 | 40.5 | 43.459 | 82.668 | 0.0 | NNC | 3.0bNNC | 2.8p | 1
| 2014 0331 0034 | 19.8 | 48.153 | -71.324 | 18.0F | OTT | 2.2N | 1
| 2014 0331 0038 | 17.9 | -39.447 | 175.748 | 18.5 | WEL | 1.6WEL | 1.8WEL | 1.6L | 1
| 2014 0331 0039 | 38 | 28.200 | 129.520 | 19.0 | JMA | 1.6 | 1
| 2014 0331 0039 | 46.8 | 36.920 | 141.420 | 35.0 | JMA | 3.0 | 1
| 2014 0331 0041 | 09.3 | 23.971 | 122.289 | 27.9 | TAP | 2.2L | 1
| 2014 0331 0041 | 42.3 | 67.896 | 20.111 | 0.0F | HEL | 1.3L | 1
| 2014 0331 0043 | 32.1 | 23.851 | 120.970 | 14.6 | TAP | 0.8L | 1
| 2014 0331 0045 | 43 | 36.680 | 140.670 | 10.0 | JMA | 2.9 | 1
| 2014 0331 0050 | 15.5 | 17.457 | -61.224 | 8.2 | TRN | 3.2D | 1
| 2014 0331 0054 | 02.5 | 36.819 | -97.600 | 5.0 | TUL | 2.7L | 1
| 2014 0331 0054 | 46.1 | -32.913 | -178.539 | 35.0F | IDC | 4.6bIDC | 4.8xIDC | 4.1xIDC | 1
| 2014 0331 0055 | 13.9 | 40.220 | 141.740 | 18.0 | JMA | 0.9 | 1
| 2014 0331 0056 | 12.2 | 43.478 | 12.407 | 8.5 | ROM | 0.7L | 1
| 2014 0331 0058 | 01.6 | 33.906 | -117.944 | 16.3 | ANF | 2.4LPAS | 2.3L | 1
| 2014 0331 0059 | 45.3 | 43.340 | 12.541 | 9.4 | ROM | 1.5d | 1
| 2014 0331 0059 | 50.2 | 33.911 | -117.944 | 16.9 | PAS | 2.3L | 1
| 2014 0331 0059 | 53.9 | 37.510 | 141.350 | 37.0 | JMA | 2.2 | 1
The source 'us' is translated into USG in the S-file but the others are left as they are, e.g. 'ak' is Alaska earthquake Center'.

The input magnitude types from the USGS text file are converted to SEISAN magnitude types as follows:

mb b
ML L
mB B
Ms s
MS S
MW W
Mc C
mw W
mww W
ww W
wr W
ml L
md C

The program has several options: Set limits for minimum and maximum depths and minimum magnitude
Set time interval for download
Run once for the given parameters from command line
Run at given time intervals to always maintain the local data base. If a new location of the same event is found, the old event is replaced.

The program has no parameter files, all options are given as options on the prompt line.

Typing get_web_locations -h the following options/info are shown (platform dependent)

SEISAN_TOP..........................: /home/seismo/snew

Valid program options, int is integer, float is a floating point number and text is an ascii string:

-h show this help info
-loop int 1-loop forever, 0-one query (0=default)
-tsrt int minutes back, start-time.  
(Number of minutes to start from current time)
-tstp int minutes back, stop-time.  
(Number of minutes to stop from current time. This should not be too close to real time in order for the data center to calculate the hypocenter. 50 minutes is a reasonable number. These two parameters are only used if the program runs continuously. The interval of looking for data is tstp-tsrt.
-mag float minimum magnitude
The minimum magnitude to be included. The magnitude is the first found in case there are several types of magnitudes. If no magnitude is given, all magnitude events are used.
5.2. GETTING DATA INTO THE DATABASE

-\text{mind} text minimum depth (default: 0)
-\text{maxd} text maximum depth (default: 800)
-\text{db} text SEISAN database name, max 5 characters
-\text{srt} text start-time, format: 2018-02-20T03:45:00
-\text{end} text end-time, format: 2018-02-22T08:23:00

Start and stop time for a one time extract. Cannot be used at the same time as \text{tsrt} and \text{tstp}.

To run the program in a continuous loop, the following parameters MUST be specified:

-\text{loop} 1
-\text{tsrt} t1 (start time, number of minutes back in time)
-\text{tstp} t1 (stop time, number of minutes back in time)

Example 1: Collecting data continuously from command line:

\text{get\_web\_locations} -\text{loop} 1 -\text{tsrt} 60 -\text{tstp} 50 -\text{mag} 3.0 -\text{db} MYDB

The program will query data from 60 minutes ago until 50 minutes back in time from now. The program will then sleep for 60 - 50 = 10 minutes and start again. Minimum magnitude is 3.0

-\text{srt} and -\text{end} cannot be specified when running in loop mode.

Example 2: "one time" data collections with no options

\text{get\_web\_locations} -\text{db} TEST -\text{srt} 2018-01-01T00:00:00 -\text{end} 2018-01-10T23:59:00

Example 3: "one time" data collection from start time to end time from command line:

\text{get\_web\_locations} -\text{mag} 5.0 -\text{db} ABC -\text{mind} 10 -\text{maxd} 100 -\text{srt} 2018-01-01T00:00:00 -\text{end} 2018-01-30T23:59:00

Example of output S-file

2018 1 1 052 31.0 D -0.280 123.700 127.2 USG 4.5b USG1
WEB event-id: us1000byph 3
LOCALITY: 80km SSW of Molibagu, Indonesia 3
ACTION: NEW 18-03-26 20:27 OP:SEIS STATUS: ID:20180101005230 I
STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO AIN AR TRES W DIS CAZ7

Example 3: Continuous mode as cron job on a Linux system

The setup below will start every hour at 0 minutes in a bash environment
0 * * * * /home/seismo/get_web_locations/cron_restart.bash 2.0 MYDB 0 800 2 1 >/dev/null 2>&1

The setup below will start every hour at 0 minutes in a csh environment

0 * * * * /home/seismo/get_web_locations/cron_restart.csh 2.0 MYDB 0 800 2 1 >/dev/null 2>&1

cron_restart.csh

#!/bin/csh
rm -f /home/seismo/get_web_locations/usgs.log
source /home/seismo/COM/.SEISAN
echo MAG: $1 DBASE: $2 MIN_DEPTH: $3 MAX_DEPTH: $4 START_HOUR: $5 STOP HOUR: $6
set start_time='date --utc '+%Y-%m-%d'T%H:%M:00' --date -$5hour'
set stopp_time='date --utc '+%Y-%m-%d'T%H:%M:00' --date -$6hour'
echo $start_time
echo $stopp_time
cd /home/seismo/get_web_locations
/home/seismo/get_web_locations/get_web_locations -mag $1 -db $2 -mind $3 -maxd $4 -srt "$start_time" "$stopp_time" > usgs.log

cron_restart.bash file

#!/bin/bash
rm -f /home/seismo/get_web_locations/usgs.log
source /home/seismo/COM/.SEISAN
echo MAG: $1 DBASE: $2 MIN_DEPTH: $3 MAX_DEPTH: $4 START HOUR: $5 STOP HOUR: $6
start_time='date --utc '+%Y-%m-%d'T%H:%M:00' --date -$5hour'
stopp_time='date --utc '+%Y-%m-%d'T%H:%M:00' --date -$6hour'
echo $start_time
echo $stopp_time
cd /home/seismo/get_web_locations
/home/seismo/get_web_locations/get_web_locations -mag $1 -db $2 -mind $3 -maxd $4 -srt "$start_time" "$stopp_time" > usgs.log

5.2.6 Database security

Duplicate ID:

Since the database consists of single files with names corresponding to time down to the second as well as the event type (L, R or D) it will sometimes happen that two events will get the same name. Thus copying in a new event with the same name could overwrite the existing event, and the user would never know. In SEISAN, from version 5.0, some security has been put in. New data can enter the database with 4 programs: SPLIT, EEV, MULPLT and AUTOREG. With all programs, the user will be prompted if a new event is about to overwrite an existing event. Both SPLIT and EEV have the possibility to create alternative ID’s if the user wants both the new and old event, while MULPLT and AUTOREG just offers the possibility to skip a double event. If a new ID is created, an attempt will be made to use a time one second later. If that also corresponds to an existing event, the next second is attempted etc. This allows for 60 events to be registered in the database with the same minute and event type. If an event has got the ID changed, the header line in the file is NOT changed, however the ID line is of course changed. This will be indicated on the ID line with a ‘d’ at the end of the ID number.
5.2. GETTING DATA INTO THE DATABASE

Deleting events:
Event here means S-file in the database. Events are only deleted when using EEV, either with the EEV delete command D or the EEV append command A. In both cases, the deleted event is stored in the DELET database before being deleted from whatever database. Even if the system contains many databases, there is only one DELET database. This means that deleted events from different databases are mixed in DELET. In order to restore an event, enter DELET database with EEV and copy the deleted event back with the C command. It is up to the user to manually clean up the DELET database.
There is one more final security. If an event has been deleted from a database, but an UPDATE has not yet been made, the event might be in the CAT part of the database and can be extracted by SELECT or the editor.

5.2.7 Data base tools, content and checks

Content of data bases, program BASE:
In the REA directory, a binary file called REA.LOG contains information about number of events in all data bases. Initially the file has no information, but each time programs EEV, HYP, UPD, CHECK_BASE or COLLECT are executed, the information is updated for the months accessed. The information can be displayed with program BASE, which first shows available data bases and the user can then select one to get info for particular months. Make sure to use right case for data base names, always in upper case on Unix systems.

Check for data base related errors, program CHECK_BASE
The data base depends on error free S-files and that there is a correspondence between the S-file name and the event ID. This should normally be ok, however errors can occur during editing or there can be program crashed producing errors. The program reads the data base and checks for:

Missing ID lines: If ID line is missing, it can be put in with command s, doing an UPDATE, using UPD or command us in EEV.

No correspondence between ID line and S-file name: A serious error has occurred. try to find out what is correct, the ID or the file name. An UPDATEi, UPD or ius in EEV cures the problem, however data might be lost.

Error in S-file: All parameters are checked and files with non standard parameters are indicated. The error can be a number in a wrong position. The errors should be corrected.
For all the above 3 cases, an index file is generated with bad S-files and EEV can the be used directly with the index file to access the bad S-files. THIS ONLY WORKS WITH ONE DATA BASE AT A TIME.

It is recommended to run check_base in case of system crash or as a security, just before an UPDATE.

5.2.8 High accuracy in SEISAN

SEISAN can use higher accuracy than the default. The goal is to have an accuracy of 1 ms in time and 1 m in location.

In order to write out the high accuracy numbers, a new parameter has been added to SEISAN.DEF. The parameter is HIGH_ACCURACY. Setting it to 1.0 enables high accuracy operation. This parameter affects the programs MULPLT, FK, HYP and UPDATE.
Station locations: The station file looks like before except that in order to get higher accuracy of station locations, the minutes of latitude and longitude are specified without the point. E.g. the minutes 22.122 can now be written as 22122 in the same columns as before while if the point is given, only 2 decimals can be used as 22.12. This changes do not affect any old station coordinates. Programs reading station coordinates, will use high accuracy input if available.

EPIMAP will always read in high accuracy mode, if any high accuracy data is present, whether station locations or hypocenters.

FK will always read high accuracy station coordinates, if available and FK can therefore now be used with very small arrays.

Programs with output affected by high accuracy mode:

MULPLT will write the phase readings as f6.3 instead of f5.2 like e.g. 11.234 instead of 11.23. For normal use, this is not needed and the files look better if high accuracy mode is not used. In new Nordic format, seconds are always written with 3 decimals.

HYP and UPDATE writes an extra high accuracy hypocenter line which has been given type H. An example is

<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th>Day</th>
<th>Year</th>
<th>Month</th>
<th>Day</th>
<th>Seconds</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Depth</th>
<th>RMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1996</td>
<td>6</td>
<td>3</td>
<td>2006</td>
<td>35.5</td>
<td>D</td>
<td>46.787</td>
<td>153.722</td>
<td>33.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>6</td>
<td>3</td>
<td>2006</td>
<td>35.511</td>
<td>D</td>
<td>46.78711</td>
<td>153.72245</td>
<td>33.011</td>
<td>1.923</td>
<td></td>
</tr>
</tbody>
</table>

The format is

Column
1 -15 As type 1 line
16 Free
17 Seconds, f6.3
23 Free
24-32 Latitude, f9.5
33 Free
34-44 Longitude, f10.5
44 Free
45-52 Depth, f8.3
53 Free
54-59 RMS, f6.3
60-79 Free
80 H

5.3 Interactive work with earthquake locations, EEV command

The idea of SEISAN for interactive work is that the user should be able to easily jump from event to event and run several different programs with one event without restarting every time. This is done with the command EEV (see below). In this interactive mode, events are picked, edited, located, moved, deleted etc. until a satisfactory solution is found. In the interactive mode, NO UPDATING of the location in the permanent output CAT directory is done since it is too easy in interactive mode to accidentally change something. The permanent updating of CAT directories can only be done for one or several months at a time (see UPDATE command) in order to ensure that nothing is forgotten within a month. Thsi also updates the S-files. Updating single events can also be done in EEV.
5.4. HOW EEV WORKS

Once the events have been updated, further work can be done (like searching for specific events or making a bulletin) using single programs which read directly from the database. Most of the analysis programs will also work without using the database structure that is e.g. searching in single file with many events. For more details of the analysis programs, see chapter 6.

5.4 How EEV works

It is now assumed that data has been entered into the database. The fundamental tool for the database is then the EEV program, which mostly works within the limits of one month in the standard database or with whatever the user has of S-files in his own directory. SE works in a similar way but with a graphical user interface. Optionally, EEV can also work with several months. A special option is to use a list of files in an INDEX file, see end of this section and SELECT program. Some of the commands available within EEV are also available within programs. See below for more details on EEV.

The EEV program reads the file names of all S-files in the database monthly directory (or local directory or index file), positions the pointer at the first event and asks for a command to be performed for the current event or to find another event. If the command is to use a program, control is handed over to that program, which on completion hands control back to EEV. In this way, many different independent programs can be used from within EEV, e.g. several different location programs can be installed.

EEV can be started in several ways:

EEV with one month in default database: EEV yyyyymm.
E.g. EEV 199201 would work on January 1992 on the standard BER database. It is here also possible to give a more precise start time like EEV 1992011520 to start with the first event at or after January 15 at 20 hrs.

EEV with one month in alternative database: EEV yyyyymm BASE. BASE is the database. To work on the NAO base, the command would be EEV 199201 NAO.

EEV with several months in default database: EEV yyyyymm YYYYMM yyyyymm is start year and month and YYYYMM is end year and month.

EEV with several months in alternative database: EEV yyyyymm YYYYMM BASE yyyyymm is start year and month and YYYYMM is end year and month.

EEV to work with events is local directory: EEV
Only the S-files in local directory will be used.

EEV to work with an index file: EEV index.out
EEV can work with an index file and the command would be EEV index.out, where index.out is the index file name (can have any name as long as it contains a ‘.’ except when used with HYP). For information on index files, see 14.

Databases can have 1-5 letter names and the user specify 1-5 letters. The real names in the directory structure are always 5 letters so if the user specifies e.g. a base name of BA, the real name will be BA____. The full 5-letter name can also be used.

The commands in EEV mainly use only one letter unless a date or a number has to be given. To get a short explanation, type ? and you will get:

-----------------------------------------------------------------------
Help on EEV, all commands are given in lower case
*** Basic commands

Q: Quit EEV
? Print this help file

*** Navigation in EEV

B: Back one event
#xx: Go to event # xx, also works without the #
Dxxxxxx: Go to first event on date xx, hour xx, min xx
Eyyyymm: Let EEV session end with year yyyy and month mm
Jyyyymm BAS: Jump to year yy and month mm in base BAS
Sxxxxxx: Search for next two events which are within xxxxx seconds. If time is blank, a default of 180 secs is used
SS: Find next unprocessed event in base

*** Event information

EP: Edit print.out file from last location
T: Type event
TT: Type only header line of event
W: Show location of waveform files listed in s-file

*** File operation like copy, delete etc.

Axx: Append event # xx to current event, original event remains
AA: Append current event to next event, original event remains
C: Copy event to another data base or to current directory
CM: Copy out several events to eev.out
D: Delete current event, you will be prompted to confirm
DUP: Duplicate current event in data base, different id
EMAIL: Send S-file as email as defined in SEISAN.DEF.
NEW: Add a new event to data base
PRINT: Print S-file on printer
R: Rename event type, must be L, R or D
PUT: Register event
REG: Register event
US Synchronize s-file name and ID with origin time and make an ID line if not there

*** File modification and edit

ARC: Add an archive line
ARC_: Add an virtual network archive line using the name given after the _ like: ARC_GSN
ARCDEL: Delete all archive lines
5.4. HOW EEV WORKS

ARX: Extract waveform file from archive and put in S-file
COMMENT: Write comment lines in S-file
COM: Same as COMMENT
COMF: Add one or several felt information lines
COML: Add one geographical location line
COMT: Write intensity comment line in S-file
DD: Duplicate header line in S-file
DELS: Delete specific phase lines in S-file, like P-phases
E: Edit S-file
EXP: Enter explosion information
FIX: Fix, unfix or set fixed depth
FIXE: Fix or unfix epicenter
FIXO: Fix or unfix origin time
IC: Insert comment as specified in SEISAN.DEF
INPUTEPI: Input hypocenter and origin time in S-file from st. input
INPUTONE: Input an additional type one line (hypocenter line)
INPUTFPS: Input complete fault plane solution line
INPUTMAG: Input 3. magnitude on header line
IM: --------------------------------
IFP: Input fault plane solution strike, dip, rake, Ag., Prog. Q.
INPUTX: Input of xnear and xfar, RESET TEST 107 must be 1.0
M: Input model indicator on header line
MTD: Delete all mt and synt lines
MTE: Edit MT parameters in S-file
STD: Set or unset start location depth flag, also set depth
STE: Set or unset start location epicenter flag

*** Hypocenter location

EP: Edit print.out file from last location
GRID: Locate by grid search
GHYP: Show Google map or Earth (or not) at each location with HYP
FIX: Fix, unfix or set fixed depth
FIXE: Fix or unfix epicenter
FIXO: Fix or unfix origin time
H: Locate with Hypoinverse
HYP071: Locate with Hypo71 (linux only)
IL: ISC location program (linux only)
INPUTX: Input of xnear and xfar, RESET TEST 107 must be 1.0
L: Locate event, will also calculate magnitude if not locatable but distance is present, Hypocenter
LR: Reject outliers and locate, see also command UR
LL: Locate current and next event together
Lxx: Locate current and event xx together
M: Input model indicator on header line
U(UPDATE): Updates S-file with hypocenter etc.
UR: Reject outliers and update S-file with hypocenter etc.
CHAPTER 5. USING SEISAN

*** Plot signals and time sequence analysis

P: Plot event, also make hard copies and pick phases
PO: Plot event with defaults, multi trace mode
POL: Plot vertical channels P onset with polarities
POO: Plot event with defaults, single trace mode
PB: Plot Bouch synthetics
PH: Plot Herrmann synthetics
PW: Plot wkbj synthetics
PM: Plot moment tensor synthetics
PD: Plot extracted data file for moment tensor, mulplt.wav
PG: Ploty Greens's functions when doing MT
SAC: Run SAC, Linux only

*** Plot epicenter

GMAP: Make epicenter map with Google Map
MAP: Make a map of current location with EPIMAP
MAPPF: Makes a map like the MAP command showing fps if there
MAPG: Makes a GMT map like the MAP command
MAPGF: Makes a GMT map like the MAP command showing fps if there

*** Plot spectra, magnitudes vs distance, wadati, picture etc

PMM: Plot moment vs time
PML: Plot Ml vs distance
PP: Plot picture file in PIC or in local dir
PS: Plot spectra and WA picks made with command AM
PSPEC: Same as PS
RMSDEP: Calculates and plots RMS as a function of depth
WAD: Make a wadati diagram
PF(IT): Get back azimuth and app. Velocity for network P-arrival, plot
TTPLOT: Make travel time plot

*** Automatic routines for picks, amplitudes and spectra

AMPRATIO: Automatic amplitudes for ratio used with fps
AR: Same as above
AUTOSIG: Automatic processing with autosig
AUTOMAG: Automatic magnitude Ml and Mw, all defaults
AM: Same as automag
AMI: Automag with questions about window lengths
AUTOPHASE: Automatic phase picking with Lomax routine in AUTOPHASE
AP: Same as autophase
Z: Auto pic current event, if readings available, new pics will be added with a flag

*** Fault plane solution, plot, manipulation
5.4. HOW EEV WORKS

F: Make a fault plane solution with FOCMEC
F_OLD: Old ------, before 2017
FH: Make a fault plane solution with HASH
FI: Make a fault plane solution with PINV
FP: Make a fault plane solution with FPFIT
FPFIT: -------------------------------------
FO: Plot all fault plane solutions, no questions, with polarities
FOO: Plot all fault plane solutions, no polarities plotted or needed
FM: Plot moment tensor solution with mopad
FD(IF): Compare P and T axis from two different fps
INPUTFPS: Input complete fault plane solution line
IFP: Input fault plane solution strike, dip, rake, Ag., Prog, Q.
MAPF: Makes a map like the MAP command showing fps if there
MAPGF: Makes a GMT map like the MAP command showing fps if there
POL: Plot vertical channels P onset with polarities

*** Moment tensor

MTP: Make Moment Tensor (MT) parameters
MTG: Make Green’s function for MT
MTI: Invert for MT solution
FM: Plot moment tensor solution with Mopad
MTD: Delete all MT and synt lines
MTE: Edit MT parameters in S-file
INVRA: Make moment tensor inversion with Invrad
PM: Plot moment tensor synthetics with original data
PD: Plot extracted data file for moment tensor, mulplt.wav
PG: Plot Greens’s functions when doing MT

*** Synthetic seismogram and travel times

BOUCH: Run Bouchon’s modeling program
BOUSEI: Make SEISAN file from Bouchon synthetic file
HERRMANN: Run Herrmann’s modelling program (not PC)
HERSEI: Make SEISAN file from Herrmann synthetic file
IASP: Generate arrival times using IASPEI91 tables
SYNT: Make parameters for synthetic modelling
WKBJ: Make synthetic seismograms with WKBJ program

*** Macroseismic information

MAC: Input macroseismic information
MACROMAP: Make a GMT based map of felt information, Unix only
PMAC: Macroseismic Windows program
EM: Edit ISO file in ISO directory
COMF: Add one or several felt information lines
COMT: Write intensity comment line in S-file
*** Explosion information

EXP: Enter explosion information
QUARRY: Quarry check command

*** Other commands

O: Operating system command, e.g. ols is ls, ocd test is cd test
do not currently work on command with prompt input like epimap and
collect
USERCOM: Start user defined command as ‘usercom -sfile <sfile-name> ’
MODELS: List MODEL.DEF file in DAT
UP: Update list of S-files in memory

*** end of file

jh sep 22 2020

Note: Command letters can be upper or lower case.

Comments to commands:

#XXX : Go to event by number. When giving a number, only give the number of digits needed, no
formatting. Thus e.g. to find event 7 or 777, write 7 or 777 respectively. If there is not an event
corresponding to the parameter specified, EEV will go back to event #1. In the number command, #
can be omitted.

Axxx: Append another event to current event. The event specified is appended to current event. All
header and lines in both files are saved and put in order in the current event. The main first header is
from the current event. The ID line for the appended event is saved as a comment line. The user will be
questioned if the appended event is to be deleted.

AA: Same as above using next event.

ARC: Adds an archive line so event can be plotted, see SEISAN.DEF for defaults to set. Use 'ARC
.VNET' (where VNET is an example of virtual network) to create ARC line with the virtual network.
ARC _ : Adds an archive line for the virtual network name given after the _ like ARC _GSN for the GSN
virtual network.

ARCDEL: Deletes all ARC lines in the S-file

ARX: Extract waveform file corresponding to S-file and adds thew waveform file name to the S-file. See
program GET_ARC for details.

AMPRATIO or AR: Run program AUTORATIO [25] to measure P and S amplitudes to be used as
amplitude ratios in FOCMEC and HASH.

AUTOSIG: Automatic processing with autosig program.

AUTOMAG and AM: Automatic amplitude for Ml and automatic spectral analysis. Only S-waves are
used and fixed windows are used. The results are overwriting result in S-file. For more details see program
AUTOMAG.

AMI: AUTOMAG, but questions about spectral window and window for MI are asked. Type of spectrum
(P or S) is also asked.
5.4. HOW EEV WORKS

AUTOPHASE and AP: Automatic phase picking with Lomax picker.

AUTORATIO and AR: Automatically calculates amplitudes from spectra and time domain for use with FOCMEC.

B: Back one event

BOUCH: Run Bouchon’s modeling program

BOUSEI: Make SEISAN file from Bouchon synthetic file

C: Copy events

There are two options, copy the event to another database given by a 1-5 letter name (upper case) or to a file EEV.OUT in your working directory. Several files can be extracted within one EEV session to the same EEV.OUT file. A new EEV session deletes the previous eev.out file. The C option can be used to recover files from the DELET database of deleted events. In addition to making the EEV.OUT file, an index file is also made called indexeev.out. THIS FILE IS NOT DELETED WHEN EEV STARTS UP since the intention is to be able to use EEV to make an index file of interesting events from several months. You can then start eev with the selected events with command EEV eevindex.out. Note: The other data base can also be a local data base “,” in which case EEV should not operate on the same local data base.

CM: Copy many files to eev.out. The copying starts at current file and the user is asked for the number of files to copy.

COMMENT: Comment are written into S-file, terminated by a blank line.

COM: Same as COMMENT

COMF: Add one or several felt information lines

COML: Add one geographical location line

COMP: Write province comment line in S-file

COMT: Write intensity comment line in S-file

DXXXXX; The D-command is used to jump to another event at a given date and time, normally only day is used: The hour can optionally be specified. E.g. d2205 will find the event nearest in time after day 22 at 05 hours. If both day and hour is used 4 digits MUST be given e.g. 0708. Highest accuracy is the nearest minute.

D: Delete event You are asked for confirmation. After the event has been deleted, all S-file names are read in again and all event numbers after the deleted event are therefore changed. The deleted event is automatically saved in the DELET database. If the event is present in the CAT file, it remains there until the next update is done, see UPDATE command in [14].

DELS: Delete specific phase lines in S-file, like P-phases. This is the simplest way to delete large specific part of an s-file.

DD: Duplicates the header line

DUP: Duplicates an event in the database. The duplicated event has an ID, which is one second different from the original event. The command can be used to split an event in two and then manually deleting phase lines in each.

E: Edit the event. As default on Linux vi is used and on PC Notepad is used. The editor can be changed,
see section 3. When control goes back to EEV, the file is checked for possible typing errors or other format problems. If a problem is encountered, the line with the problem is displayed with an indication of where the mistake might be, and the user is returned to the editor. Alternatively the error can be ignored. The file is also checked for missing ID and consistency between file name and ID. Problem: Some editors will keep a backup copy of the original file so 2 files might be present with one e.g. with the additional extension .BAK. EEV (from version 7.2) will only use the original file, but there is no check on what backup files might accumulate.


EMAIL: Send s-file as email using mailx program defined in SEISAN.DEF (variable MAILX COMMAND) (Linux only) for the addresses defined in SEISAN.DEF (variable ALERT EMAIL).

Eyyymmm: Giving this command will make the current EEV session end with year yyyy and month mm within the same data base. When EEV gets to the end of the month, pressing return will move EEV to the first event of the following month instead of to the first event of the same month.

EP: Edit the print.out file, useful to see what happens if something wrong with location or no location.

EXP: Input of explosion information. This command creates 3 new lines (see format description in Appendix [A] and changes the main header line event type to explosion (E). The user is asked for location, time, charge and comments. The explosion agency is used to classify types of sites and can be used by SELECT for searching. If no event is available, a new event must be created with EEV command NEW.-

F: Make a fault plane solution. The program uses polarities. See section 23.1 for more details.

FOLD: Use the old version of Focmec, before SEISAN version 10.6.

FO: Just plot a solution in S-file. It must have data so it can be located. Polarities are shown.

FOO: Plot solutions even if no phases in S-file. Polarities are not plotted.

FD: Calculate the difference in orientation of P and T axis of any two fps

FI: Fault plane solution using PINV

FH: Fault plane solution using HASH.

FP: Fault plane solution using FPFIT.

FM: Plot moment tensor solution with Mopad.

FQ: Enter quality of fault plane solution. It is also possible to order the solutions so the prime comes first.

FIX: Fix or unfix depth. It is also possible to give the depth.

FIXE: Fix or unfix epicenter

FIXO: Fix or unfix origin time

GMAP: Make an epicenter map of current event using Google Map. It is also possible to make maps with many epicenters using GMAP outside EEV, see section 10.5 for more details.

MAPG: Makes a GMT map like the MAP command using the GMT software.

GHYP: Show Google map or Earth (or not) at each location with HYP. The Google Earth option requires Google Earth to be installed and in the path.

GRID: Hypocenter is started up and will ask for the grid: Latitude and longitude range and grid spacing. A maximum of 71 points can be used in each direction. The point with the lowest RMS is found and
the corresponding location and residual is printed on the screen. It is now optionally possible to plot the contours on the screen. The map coordinates used are as defined in SEISAN.DEF. Note that the grid search is using exactly the same parameters as Hypocenter. This includes all weights and phase types. The depth is fixed to the depth given in the S-file header line. For more details and an example, see application note epi.pdf in INF.

H: Locate with Hypoinverse, no database update is made, no Nordic output format file.

HERRMAN: Herrmann’s modeling programs, only on Sun, might work on Linux, not tested.

HERSEI: Make a SEISAN waveform file from output of Herrmann modeling, only tested on Sun.

HYPO71: Locate with HYPO71. The database is not updated (not well tested on PC).

IASP: Generate a file with theoretical arrival times for the current event for stations listed in S-file. The command will only work if the event has an epicenter and origin time in header line or a subsequent type 1 line, see also HERSEI: Make a SEISAN waveform file from output of Herrmann modeling, only tested on Sun.

HYPO71: Locate with HYPO71. The database is not updated (not well tested on PC).

IASP: Generate a file with theoretical arrival times for the current event for stations listed in S-file. The command will only work if the event has an epicenter and origin time in header line or a subsequent type 1 line, see also INPUTEPI and INPUTONE. These theoretical times will then be displayed with mulplt, the next time command P is used in EEV. The theoretical times are listed in file iasp.out. See section 30.4 for more information. The command can also be used directly from MULPLT and in this case theoretical travel times are calculated for all stations in waveform file(s).

IC: Enter one of predefined comments. The comments are given in SEISAN.DEF and any number of comments can be predefined. Example:

```
7 1 Jan 2015 13:35 39 LQ 78.312 7.560 15.0 0.9 2.0LBER 8 ? ic
```

Predefined comments are:

1 Calculated depth is 0km, fixated to 10km due to uncertain depth
2 Calculated depth is 31km, fixated to 10km due to uncertain depth
3 Calculated depth is 31km, fixated to 10km due to uncertain depth
4 Probable explosion due to spectrogram and waveform
5 Induced event reported on lkab.com,
6 Probably explosion at Rana Gruber, Storforshei
7 Probably explosion at Titania
8 Probably explosion at Norsk Stein
9 Probably explosion at Slovaag, Sogn & Fjord
10 Probably explosion at Eikefet quarry

enter number
CHAPTER 5. USING SEISAN

IL: Makes a location with the ISC location program. For more info, see section 7.3

INPUTONE: Makes an additional type one line (hypocenter line) in the file. Enter the data exactly under the columns indicated. The line will be entered exactly as written, so it is possible to enter any part of the information.

INPUTEPI: Works like INPUTONE, except that it overwrites information on the first header line if non-blank information is given. Use INPUTEPI to add information to the first header line like e.g. the depth. If existing nonblank characters on the line are to be replaced by blanks (e.g. remove a magnitude), use underscore “_”.

INPUTX: Makes a comment line with xnear, xfar and start depth values. Note that RESET TEST(107) must be set for this option to work.

INPUTFPS: Input complete fault plane solution line

IFP: Input fault plane solution strike, dip, rake, agency, program used and quality of solution.

INVRAD: Runs the moment tensor inversion program, see section 23.7.

Jyyyyymm BAS: This command makes it possible to change month and database during an EEV session by giving a new year yyyy and month mm and optionally a new database BAS. If no database is given, the same database is assumed.

L: Locate event with HYPOCENTER (same as HYP). The location does not update the S-file.

LR: Reject outliers and locate. Outliers remain in file. Can be removed with command UR. see HYPOCENTER section for more details.

Lxx: Locate current event with event number xx. This is used to check if two events belong together.

LL: Locate current and following event together.

M: Input or remove model indicator on header lien.

MAC: Enter macroseismic information, you will be prompted for all information. For details of the type of information, see definition of Nordic format, Appendix A. See also command PMAC.

MACROMAP: Felt information is read from a file with macroseismic information and plotted with GMT. The file name of the file with macroseismic observations is given in the S-file.

MAP: Start EPIMAP program to produce a map of current location. If a location is given in the S-file, this location is plotted, otherwise the event is located if possible and the resulting location used for plotting. The parameters for generating the map are set in the SEISAN.DEF file (see 3.12).

MAPF: The same as MAP but in addition the first fault plane solution found in file will be plotted. There is no requirement that there are polarities.

MODELS: Lists MODEL.DEF file in DAT that assigns names to single characters in STATIONx.HYP file.

MTP: Make Moment Tensor (MT) parameters, input wav file must be made first.

MTG: Make Green’s function for MT. MTP first.

MTI: Invert for MT solution. MTG first.

MTD: Delete all mt and synt lines.

MTE: Edit MT parameters in S-file. A menu comes up and different parameters can be edited.

NEW: Creates a new event in the database. The user is asked to give date and time and the event is created in the current monthly database.
5.4. HOW EEV WORKS

O: Give a command to the operating system. This is a very useful command, since it is possible to do almost anything without leaving EEV, including starting a new session of EEV. E.g. the command ols on Sun and odir on PC would make a directory listing. The name and path of the current s-file is copied to a file named `eev.cur.sfile`, this makes it easy to write your own programs to handle seisn data.

PB: Plot Bouch synthetics, automatic with MULPLT
PH: Plot Herrmann synthetics, 
PW: Plot wkbj synthetics, 
PM: Plot moment tensor synthetics
PD: Plot extracted data file for moment tensor, mulplt.wav
PG: Plot Greens’s functions when doing MT

PF or PFIT: Calculate the apparent velocity and back azimuth using the P-arrival times stored in the S-file. The calculation is done by a free standing program PFIT, which also can be called outside eeve. It is assumed that the arriving wave can be approximated with a plane wave so this option is intended to be used with events which are far away relative to the size of the network which then can be considered a seismic array. The station coordinates are taken from the default station file and there is no correction for station elevation. When starting the pfit option, the user will be given a choice of reference station and maximum distance from the reference station. Linear distances will then be calculated from the reference station and possible results will be associated with the reference station. All P-phases given as P, Pn, PN, Pg, PG, PKP, PB and Pb will be used and it is up to the user to ensure that the event file only contains the phases to be used. The output is displayed on the screen and the linear fit can be shown on a plot, which also can be used to interactively check individual station values, see example below.

Example run of PFIT

Give reference station, enter for using first station in list
HYA
Maximum distance from reference station, default is 1000 km

2007 1 5 1652 23.3 D 55.991-159.065 13.5 BER 19 1.2 5.3SBER 5.5BBER 5.4BPDE

Stations available: 18 Stations used: 14

<table>
<thead>
<tr>
<th>Stat</th>
<th>Delta t</th>
<th>Latitude</th>
<th>Longitude</th>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYA</td>
<td>0.000</td>
<td>61.165</td>
<td>6.182</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>LOF</td>
<td>-44.970</td>
<td>68.132</td>
<td>13.541</td>
<td>394.668</td>
<td>774.733</td>
</tr>
<tr>
<td>NSS</td>
<td>-23.720</td>
<td>64.531</td>
<td>11.967</td>
<td>310.275</td>
<td>374.227</td>
</tr>
<tr>
<td>MOL</td>
<td>-9.540</td>
<td>62.570</td>
<td>7.547</td>
<td>73.220</td>
<td>156.192</td>
</tr>
<tr>
<td>DOMB</td>
<td>-6.840</td>
<td>62.073</td>
<td>9.112</td>
<td>157.158</td>
<td>100.965</td>
</tr>
<tr>
<td>ASK</td>
<td>0.810</td>
<td>60.472</td>
<td>5.201</td>
<td>-52.618</td>
<td>-77.095</td>
</tr>
<tr>
<td>RUND</td>
<td>4.060</td>
<td>60.414</td>
<td>5.367</td>
<td>-43.680</td>
<td>-83.581</td>
</tr>
<tr>
<td>BER</td>
<td>4.820</td>
<td>60.384</td>
<td>5.335</td>
<td>-45.414</td>
<td>-86.880</td>
</tr>
<tr>
<td>EGD</td>
<td>3.270</td>
<td>60.270</td>
<td>5.223</td>
<td>-51.403</td>
<td>-99.501</td>
</tr>
<tr>
<td>ODD1</td>
<td>5.600</td>
<td>59.911</td>
<td>6.627</td>
<td>23.891</td>
<td>-139.457</td>
</tr>
<tr>
<td>KONO</td>
<td>9.630</td>
<td>59.649</td>
<td>9.598</td>
<td>183.221</td>
<td>-168.571</td>
</tr>
<tr>
<td>SNART</td>
<td>18.280</td>
<td>58.339</td>
<td>7.210</td>
<td>55.130</td>
<td>-314.293</td>
</tr>
</tbody>
</table>
CHAPTER 5. USING SEISAN

Figure 5.1: The linear fit of P-arrival times to a plane wave. For more details, see Havskov and Ottemöller [2010].

Reference station is: HYA

Back azimuth = 358.8  Apparent velocity = 19.81  corr = -0.99  rms = 0.03

Relative to the reference station, the above output gives relative P-times and relative x and y-coordinates (km). It is also seen that only 14 station were available within 1000 km from the reference station HYA. These results are also available in an output file array.out. See also array processing section 6.29 on FK-analysis.

PMM: Plot moment as a function of time. The values are taken from the SPEC lines. The intention is to get an idea if the attenuation is correct by seeing the trend. If the line has a negative slope, then the Q value is too and vice versa. On top of the plot the corresponding Q correction needed to make the moment distance independent is shown. It is calculated at a third of the corner frequency. This value is only an indication. On the plot it is possible to click on a symbol to get the corresponding channel used.

PML: Plot Ml as a function of distance. The plot can be used to judge any distance biases in the magnitude calculations. On the plot it is possible to click on a symbol to get the corresponding channel used.

O PUT: Register event. This option is mainly meant to be used with the SEISNET data collection.
5.4. HOW EEV WORKS

system. The command cleans up the S-file for all SEISNET operations. It removes commented out ID-lines and copies the waveform files given for the event from the current directory to WAV. The command is equivalent to the register command in MULPLT. If events are auto registered with AUTOREG, the command can be used to clean up and inspect incoming data without using MULPLT directly.

PMAC: Windows only program PROMAC for processing macroseismic information to calculate intensities from felt information and model the macroseismic intensities. The program can also plot associated pictures (in directory PIC). All information is stored in the S-file. The program was written by Bladimir Moreno, and has a separate manual, see INF directory. Program must be installed separately, zip file in SUP.

P: Plot event with MULPLT

PO: Use MULPLT with defaults. This means that no questions will be asked and the plot appears in multi trace mode with default channels and default filters as given in the MULPLT.DEF file in DAT. Useful option for routine inspection of raw data.

POO: Same as PO with the difference that the user enters MULPLT single mode directly.

POL: Plot vertical channels P onset with polarities using program PLOTPOLARITY The polarity picks can be changed.

PP: Plot picture files with file names stored on type P-lines. The picture files must be in working directory or PIC directory. Several files can be defined in the S-file and the user will be prompted for which one to plot. The system command used for displaying the file must be defined in SEISAN.DEF, parameter PLOT_PICTURE_COMMAND bitmap types and PLOT_PDF_COMMAND for PDF files. An example is found for test event JUNE 3, 1996 at 19:55.

PRINT: The current S-file is printed on the default printer, to set up printer command, see SEISAN.DEF (section 3.12).

PS: Plot spectra and WA picks made with AM command. The bad fits that the user wants to deselect can be deleted interactively.

Q: Quit EEV

QUARRYCHECK: Calculate distance between epicentre and quarry locations given in quarry.dat in DAT. If distance is less than maximum distance xx as given in quarry.dat, the quarry is listed. The file looks like this:

50.

50.

The number 50 is the maximum distance.

REG: Same as PUT.

R: Rename event type - Giving an event a new type requires changing the header in the S-file and the S-file name. All this is done with R-command. You are prompted for a new type (can be the same in which case nothing is done). A new S-file is made and the old deleted. The CAT-file is NOT changed so
if no UPDATE is done, the event there will remain with wrong type. Event types are L: Local event, R: Regional event and D: Distant event. Change events id - By adding a second character to the event type the event id will be changed too. E.g. changing the event to a local explosion one must type LE. Use LB to replace the E with a blank. Standard event id’s are: E = Explosion, P = Probable explosion, V = Volcanic and Q = Confirmed earthquake.

RMSDEP: Calculates and plots RMS as a function of depth for current event. Note: Program starts by reading STATION0.HYP so if current events uses e.g. STATION1.HYP, STATION0.HYP must be there also.

SAC: Convert all data to SAC format and starts the SAC processing system (not distributed with SEISAN, must be obtained separately), not on PC.

Sxxxxxx: Search for next pairs of events separated in time by xxxxxx secs (max 999999). If no value is given, 180 secs is used. The command is intended for finding events to be merged after putting together two different data sets with SPLIT. If a new time instead of the default 180 is entered, it will remain in effect for the whole EEV session. NOTE, that the search starts with the current event, so after using S, one return to go to the next event must be given to start a new search.

STE: Set or unset the start location flag for epicenter.

STD: Set or unset the start location flag for depth. The dept can also be entered.

SYNT: Make parameters for synthetic modelling, similar format as used for MT.

SS: Find next unprocessed event in database. Events, which have status in ID line as follows: SPL: split with SPLIT program, HYP: auto-located with HYP, NEW: new event from EEV or ARG: registered by AUTOREG. The idea is that when new unprocessed data have entered the database by one of these programs, it should be easy for the operator to find the event. In EEV, an N near the end of the prompt line indicates an event with this status.

T: Type event.

TT: Type only header of event.

TTPLOT: The program reads P and S-arrival times from S-file and makes a travel time plots. The program is useful for checking readings, see section 30.4. The lines connect the computed first arrivals for P and S, respectively.

UPDATE or just U: Updates (overwrite) S-file with hypocenter, magnitudes, residuals etc. Note that the CAT file IS NOT UPDATED. This can only be done with stand-alone command UPDATE, see section 14.

UP: Update EEV event list. All S-file names are read in again. Is useful if data arrives during an EEV session, like when using Copy command from another data base.

UR: Reject outliers and update S-file with hypocenter etc. Outliers are removed. See Hypocenter section for more details.

US: Synchronize S-file name and ID with origin time. Will also make an ID line if not there.

USERCOM: Starts user defined program with command usercom -sfile <sfile-name>, where usercom is the command name. This command is useful for example if you want to start your program to create a report based on the S-file, from EEV. Note: the usercom is not a SEISAN program.

W: Check if event has waveform files. If so, check in which directory they are if present on the system. The search will start in current directory, then WAV followed by all directories defined with keyword WAVEFORM_BASE in SEISAN.DEF in DAT.
WAD: The program reads the data for the event and then asks if all phases are going to be used or only phases of the same type like Pg and Sg. Ideally, only phases of the same type should be used, however in practice it might be interesting to see all data, it might give an idea about phase identification. The Wadati parameters will now be calculated and shown on the screen. Optionally a plot can now be made. The plot shows the Wadati diagram. On the left is shown all stations with corresponding S-P times. Any station on the plot can be identified with the cursor. Point the cursor near a symbol and click and the station data will be shown in the upper right hand corner. This facility is used to identify bad picks. The plot output file is called `wad.plot.eps`.

WKBJ: Make synthetic seismograms with WKBJ program. See program for details.

Z: Automatic phase picking. A waveform file must be present. See also the AUTO program section [21]. Below is shown a session with EEV on PC.

Example of using EEV for November 1993

```
eev 199311

1993 11 Reading events for base AGA 18
# 1 2 Nov 1993 17:06 48 L 60.443 4.512 2.0 1.5 N 1.8CBER 6 ?
# 2 5 Nov 1993 22:37 21 D 1 ?
# 3 5 Nov 1993 22:37 23 D 1 ?
# 4 5 Nov 93 22:39 2 L 1 ?
# 5 5 Nov 93 22:40 58 L 1 ?
# 6 7 Nov 1993 23:40 43 L 67.837 20.059 15.0 0.7 2.5CBER 7 ?
# 7 7 Nov 1993 23:43 17 L 66.307 6.919 31.0 1.4 3.1CBER 8 ? 17
# 8 7 Nov 1993 23:44 29 D 70.069 139.780 .1 0.1 7 ?
File name: \seismo\REA\AGA__\1993\11\19 0145-29D.S199302
1993 1119 0145 29.0 D 70.069 139.780 .1 BER 7 .1

19 999.9 821.9999.9 .3206E+06 .2536E+07 .2639E+08E
93111901.K41
93 1119 153 6.5 D
9311 19 0153 065.NSN_09 6
STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO SNR AR TRES W DIS CAZ7
KBS SZ EP 151 54.8
TRM SZ EP 153 03.0
MOL SZ EP 153 50.51
ASK SZ EP 154 04.0
BER SZ EP 154 05.6
EGD SZ EP 154 06.5
KONO BZ EP 9 153 49.21

# 17 19 Nov 1993 01:45 29 D 70.069 139.780 .1 7 ?
# 18 21 Nov 1993 01:53 56 L 60.184 4.965 15.0 N 0.5 2.6CBER 11 ?
1993 11 Reading events for base AGA 18
1 2 Nov 1993 17:06 48 L 60.443 4.512 2.0 2.2 1.8CBER 6 ? q
```

In the above example (PC), the month has 18 events. For each event, vital information is displayed:
Date, type, hypocenter, RMS, first magnitude and number of stations (number in S-file which might be larger than number used for location as given in S-file header line after a location). In this way the user can quickly search for events wanted and get important information without looking at all the details. The first event in the list is newly entered into the database as indicated with the N near the end of the line. In the above example, a return was made to go to next event until event #7 after which a jump was made to event 17. For this event, all parameter data was displayed with the 't' command. A return was made to event 18, another return and the event list was read in again and event #1 again became the current event. Note that not all events had a location.

Below are shown examples of the commands (C)opy, (D)ate, a(S)sociate and (A)ppend. Comment are preceded by '! ' and written in bold. The database is EAF. The example use the old Nordic format.

<table>
<thead>
<tr>
<th>Event</th>
<th>Date</th>
<th>Time</th>
<th>Type</th>
<th>RMS</th>
<th>Magnitude</th>
<th>Station</th>
<th>Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 May 1994</td>
<td>1:18</td>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1 May 1994</td>
<td>11:37</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1 May 1994</td>
<td>12:00</td>
<td>D</td>
<td>36.607</td>
<td>68.449</td>
<td>15.0</td>
<td>2.4 ! go to day 20 ? d20</td>
</tr>
<tr>
<td>366</td>
<td>20 May 1994</td>
<td>5:2</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copy event: Other database, give 1-5 letter name

Working directory in file eev.out: return

549 27 May 1994 9:27 41 L Associated
548 27 May 1994 9:27 1 L ! append to next event ? aa
Event # 549 appended to event # 548 Appended event still present
Do you want to delete appended event(y/n=return)y ! delete appended event

Backup copy saved as: \seismo\REA\DELET\1994\05\27 0927-41L.S199405 ! del. ev. save
Deleted file \seismo\REA\EAF_\1994\05\27 0927-41L.S199405 ! app.ev. del.

Stop Program terminated.

When the interactive location is finished, the database should be updated, see section 7.
5.5. WORKING WITH CATALOGS

Using EEV on a subset of events or using alternative databases:

Since the EEV procedure or the HYP program will work on an index file, the user can create a subset of his own interesting events to work with by creating his own index file with just these events. The index file can be created by searching through the database using SELECT or it can be created manually with the C-command in EEV.

Local database:

If data is extracted by using the COLLECT or SELECT and then split up again using SPLIT, it is possible to keep all files in a working directory by not specifying database when splitting up. Another simple way is to use the Copy function in EEV and copy directly from a named data base to the local data base. Programs will then look for S-files in the current directory instead of in the database.

In addition to working with index files, there is also the possibility of storing data in different databases. By default, the data is always stored in the default data base set by DEF_BASE, here BER. However, the user can also create another database structure (file structure) with another name and programs and procedures will work on that database too. There are some restrictions: The new database, which is a subdirectory under SEISMO/REA, just like BER, MUST have a 1-5 letter name. Currently, the alternative database is used in our Institute to store data from other agencies like NAO, which in some cases are copied to our own database (C-command under EEV). The name DELET is reserved for the DELET database, which is always present.

5.5 Working with catalogs

It is often convenient to have multiple solutions of hypocenters in the database S-files or the CAT-files. Typically data has been entered from different sources and merged to form a single catalog. The first hypocenter line in the file is then considered the prime hypocenter estimate and this is the one used by e.g. EPIMAP to plot the hypocenters. The order of the hypocenters can be rearranged by CAT_AGA. Several programs use all the hypocenter lines. The magnitude correlation program will search any hypocenter line and the database selection program SELECT will optionally also use all the hypocenter lines. When the data base is updated with a new location and magnitude (UPDATE, section [14], it is only the first hypocenter line which is overwritten. If there is a magnitude in the 3 position, it is left unchanged unless it has the same agency as used for updating. This is useful in normal observatory practice, where it is common to put in some external agency magnitude which then must be left unchanged. If more magnitudes than 3 are calculated, they will be placed on a subsequent hypocenter line identified by having the same year, month, day and hypocenter agency as given on the first line. In order to merge different catalogs, it might be an advantage to put all the data into a complete database where each event is one file, even when only hypocenters are available. This is done by first splitting up the catalogs with SPLIT and then using EEV to merge the events. Since there is no requirement for monthly directories to have data, this methodology can also work for historical catalogs. The data can then subsequently be put into the CAT database without relocation using the UPD command.

5.5.1 Explosions in SEISAN

Many catalogs are contaminated by explosions and in SEISAN, explosions can be dealt with in several ways. In the data base, confirmed explosion are marked with E and probable explosions with P. These
CHAPTER 5. USING SEISAN

indicators are mostly put in when the operator first registers the event. However, there is also a possibility
to automatically identify events which are probable explosions. This is done with program EXFILTER
(section 37). In the data base S-files, there is a special format for recording explosion information
(command EXP in EEV). The explosion site there can be assigned a three letter code, which can be used
by SELECT to find explosions from specific sites. In this format it is also possible to store the explosion
charge and explosion location and origin time separately from the calculated location and origin time. In
EEV, the command QUARRYCHECK, will look for known quarries near the epicenter in the S-file.

5.6 Printing

All SEISAN programs, that produce graphical output, also generate Postscript files with the file suffix
eps (note this was plt before version 8.1). These can be directly sent to a Postscript printer. It seems
that programs like Microsoft Word don’t like the SEISAN Postscript and you will need to convert your
files to another Postscript, this can be done for example with the program ghostscript using pswrite as
output device.

Note: On Solaris 7, both the lpr and the lp command for sending files to the printer, don’t create a copy
of the file before sending it (bug in Solaris). This means that a plot file can be overwritten before being
sent to the printer. Therefore when SEISAN on Unix is sending plots, the system waits for 5 seconds
after a file is sent to the plotter before continuing. This is most important when plotting continuous data
or a large number of files with MULPLT.

5.7 General Work with SEISAN

Once data is in the database and the routine analysis has been finished by running UPDATE (final
epicenters recorded in CAT and the S-files), it is possible to go on with general work with the data.
This means searching the database, making a bulletin or plotting the epicenters. It is also possible to
use some of the more specialized tools of SEISAN which include working on subsets of data or creating
other databases, see 5.4. For general use, the basic philosophy is that the user should not enter the
REA directories. All commands and programs should be used from the user’s own directory or the WOR
directory. To access part of the main database, the programs always ask for start and end date as follows:

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>19880602011001</td>
<td>including from or to the second</td>
</tr>
<tr>
<td>19880602011010</td>
<td>including from or to the minute</td>
</tr>
<tr>
<td>19880602012001</td>
<td>including from or to the hour</td>
</tr>
<tr>
<td>19880602013002</td>
<td>including from or to the day</td>
</tr>
<tr>
<td>19880602014006</td>
<td>including from or to the month</td>
</tr>
<tr>
<td>19880602015008</td>
<td>including from or to the year</td>
</tr>
<tr>
<td>BLANK</td>
<td>only used as end date, means to end of month</td>
</tr>
</tbody>
</table>

Note that the end time is inclusive, this means that e.g. 198806 includes all of June 1988.
Thus most programs will work from any given date-time to any other given date-time. Programs that
work directly on the S-files in the database (e.g. COLLECT) can work with any time interval in which
the database structure has been created. THERE IS NO REQUIREMENT THAT THERE IS DATA
IN THE INDIVIDUAL MONTHLY DIRECTORIES, ONLY THAT THEY EXIST. There are usually 4
options for database, either the standard base (often by default), the user’s own subset of the standard
base (an INDEX file or S-files in local directory) or another database. If the user has his own database
specified by an INDEX file, the event ID’s must be in that INDEX file. Since the index file gives complete
file name of event files, the index file can work on a subset of the main database.

Note that most of the programs are used as stand-alone programs, disregarding the database structure.
If one for example prefers to have all events gathered in one file rather than split into many files and
directories most programs will therefore work.

### 5.8 Graphics in SEISAN

Most programs in SEISAN producing graphics on the screen use the SEISAN graphics system (see also
chapter [44](#)). This produces fast and low quality graphics both on the screen and similar PostScript
output files. Most of these plots are not suitable for publication and many programs therefore also create
output ASCII files of the main results, which then can be put into more professional plotting routines.
The GMT (Generic mapping Tools) system is one of the more widely used plotting systems used in
seismology. Several programs in SEISAN therefore produce output that can be used with GMT or makes
plots directly in GMT. From version 8.0 of SEISAN, a script is included (GMTXY, manual in INF) which
will produce nice xy-graphics from specially made output files. So far only programs SPEC, CATSTAT
and LSQ produce these output files, the intention is to include this feature in more SEISAN programs.

If there is a need to produce better quality graphics there are several possibilities:

- **Maps:** MAPG, map with GMT, GMT must be installed, works on all platforms.
- **W_EMAP:** Windows based mapping system
- **Seismograms:** TRACEPLOT (GMT based)
- **XY-plots:** GMTXY (GMT based)
- **Maps:** create GMT input files with SEIGMT
- **Volcanic event distribution:** VOLCSTAT

Pregenerated graphics files

Files of the type e.g. png can be stored in the PIC directory or in working directory. The file names
can be listed in the S-file in type P-lines and then plotted directly from EEV with command PP. The
command to display picture files on a particular system must be defined in SEISAN.DEF.

### 5.9 Logging in SEISAN

This part describes how logging is handled in SEISAN. Logging is turned off by setting SEISAN_LOGGING
to 0.0 in SEISAN.DEF.

Logging is performed by:

- **EEV**
- **MULPLT** continuous

**EEV:**

Logging in EEV is done with respect to each event:

- When comment is inserted (COM command)
• When event is registered (REG or PUT command)
• When event is updated (UPDATE command)
• When event is duplicated (DUP command)
• When event is moved or copied to another database (C command)
• When event type is changed (R command)
• When sfile has been edited (E command)
• When event is deleted (D command)

For each event a log file is created in the LOG folder in the database folder. The log file name is linked to the sfile name, like:

S-file: /seisan/REA/TEST_/1996/06/06-0648-30R.S199606
Log file: /seisan/REA/TEST_/LOG/1996/06/06-0648-30R.S199606.LOG

To see the content of the current log file type log in EEV.
The log file contain information on when a change was done, who the operator was and what the action was. The sfile is listed in the log file to show changes.

MULPLT cont mode:
Logging in MULPLT cont mode is described on page 149, the log files are stored in the LOG folder in the database folder.

5.10 Known problems in SEISAN

This list describe some of the known problems in SEISAN:

1: Azimuthal of arrival (h command in MULPLT single trace), channels with orientation code 1 and 2 are handled as N and E. If the orientation of 1 and 2 is off North and East the estimated azimuth will be wrong.

2: Particle motion (PartM or P command in MULPLT multi trace) will not work with channels that has orientation code 1 and 2.

3: If waveform data from a three component sensor is rotated it is assumed that the response is the same on all three rotated channels (Z,R,T).

4: If MULPLT gives the following error: "--fonts-- ERROR: Unable to get font" on Linux, it means that the default font used in SEISAN is not installed.
The solution is to install the default font -adobe-courier-bold-r-normal--12-120-75-75-m-70-iso8859-1 or to use an other font
Linux users can check which fonts they have installed with the command textttxlsfonts
See FONT under SEISAN.DEF see page 31.
4: AUTOREG crash with the message
   Fortran runtime error: Bad value during integer read
   This is because the waveform filename have characters in the filename where AUTOREG assumes
   an integer, like:
   2019-03-10-1819~M.DATA, solution is to rename file to:
   2019-03-10-1819-00M.DATA

Other known problems are listed in the index.
Chapter 6

Description of Programs and Commands

This section gives user manuals for programs and command procedures used with SEISAN. Not all are as detailed as one could want, however many questions from programs should be self-explanatory. Most programs will produce output files with the extension .out and proceeding by the name of the program. E.g. output from collect, will be collect.out. Running a program twice will erase the earlier output files. If these files are to be used later, remember to rename them before running a program again. There are several programs, which have separate manuals in the INF directory.
Chapter 7

Hypocenter location programs:
HYPOCENTER, HYPO71 and HYPOINVERSE

7.1 The hypocenter program, HYP

The hypocenter program is a modified version of HYPOCENTER [Lienert et al., 1986; Lienert, 1991; Lienert and Havskov, 1995]. The main modifications are that it can accept more phases, locate teleseismic events and use input in Nordic format directly from the database. A detailed manual (earlier version, hypocent.pdf) and some of the later changes (hypocent_latest.pdf) is given in INF directory. The input parameter file with station coordinates, model etc. is STATION0.HYP, see later.

7.1.1 Phases

Local crustal phases:

The program will accept P, Pg, Pn, S, Sg, Sn, Pb, Sb, Rg, T and Lg phases and when locating teleseismic events most of the IASPEI phases (see below). If only P or S is given, the fastest phase is used as in the original version of the program. The phase used by the program is indicated in output, see later.

7.1.2 Azimuth and single station location

The program also uses observed station azimuths as given in the Nordic Format. Station azimuths can be obtained with either 3-component stations or array stations or by using a local network as an array (see EEV pfit option) This means that the program can locate with one station if it has at least two phases like P, S and azimuth. Azimuth residuals contribute to the overall rms, see TEST(52) and section on weight. In order to locate with one station, azimuth and P and S, TEST(56) MUST be set to 1. Note that the depth then will be fixed to the starting depth. So if the starting depth is larger than the hypocentral distance, no solution is possible and the starting depth must be set to a value smaller then
the hypocentral distance. This can be done in the STATION0.HYP file or individually in the S-file. Known problem: If Azimuth on one station and P and S on another station, HYP might not locate properly.

NOTE: XNEAR must be larger than the distance to the station. If TEST(58) is set to a value lower than the apparent velocity associated with the backazimuth, no location will be made.

### 7.1.3 Magnitudes

The phase readings for magnitude determination is described on page [73](#).

In SEISAN version 8.3, there are substantial changes in the way amplitudes are read and two new magnitude scales have been added (broad band body and surface wave magnitudes). Furthermore, the Richter attenuation curve is now used be default for the body wave magnitude. The phase names used for amplitudes have also changed. These changes are due to the new standards for magnitude calculation approved by the IASPEI. For more on the application of the different magnitude scales, see Havskov and Ottomöller [2010].

Magnitudes are calculated using coda, amplitude and spectral level. Parameters are given in the station file using the RESET TEST variables. For magnitude based on amplitude, the amplitude must be given in nanometers or nm/s in the input file (SEISAN standard).

**Local magnitude Ml**

The formula used to calculate local magnitude is

\[
Ml = a * \log_{10}(amp) + b * \log_{10}(dist) + c * dist + d
\]

where \(a, b, c, d\) are constants, \(\log_{10}\) is logarithm to the base 10, \(amp\) is maximum ground amplitude (zero−peak) in nm and \(dist\) is hypocentral distance in km (RESET TEST 75-78). The default constants used, if not set in STATION0.HYP, are for California [Hutton and Boore, 1987] which gives the following relation

\[
Ml = \log_{10}(amp) + 1.11 \log_{10}(dist) + 0.00189dist - 2.09
\]

It is here assumed that the gain of the Wood-Anderson instrument is 2080. An amplitude of 1\(mm\) of the Wood Anderson seismogram is then \(10^{6}nm/2080\) and inserting this amplitude above together with a distance of 100 km gives magnitude 3 as originally defined by Richter. If using the default STATION0.HYP, distributed with SEISAN, the Ml parameters are for Norway. It is assumed that the maximum amplitude is picked on a seismogram simulating the original Wood-Anderson seismogram, see program MULPLT. SEISAN uses hypocentral distance, while the original Ml scale used epicentral distance (no deep earthquakes in California). We use hypocentral distance so Ml also can be used for deep earthquakes, but the user should be aware that the Ml relation for deep earthquakes might be different from the relation for shallow earthquakes.

Local magnitudes are only calculated for events with epicentral distance LESS THAN TEST(57) (default 1500 km) and if the period is less than 5.0 secs. All amplitudes for the phases ‘L’, ‘S’, Sg, SG, AMP, and AML, AMPL, IAML or blank are used. This means that if an amplitude is picked on both Lg and Sg, both will be used. The period is not used. The many possible phase names is a result of changes over time and thus to ensure that Ml is calculated correctly with older data. From version 8.3, MULPLT produces the standard IASPEI name IAML.

**Local magnitude Mn**
The MN scale has recently been defined for the North Atlantic region [Kim and Ottemöller, 2017]. This scale is based on Pn and Sn amplitudes measured, just like ML amplitudes, on the Wood-Anderson simulated trace. The reason for introducing MN was that in many regions ML is defined to correct for attenuation in continental crust where Sg/Lg are typically the phases with largest amplitudes. However, these are not propagating in Oceanic crust and amplitudes need to be measured from Pn or Sn phases.

The MN scale in SEISAN is implemented as described in [Kim and Ottemöller, 2017], given by

$$MN = \log_{10}(amp) - a \log_{10}(100/dist) + S + E + b$$

where the amplitude is in units of nm and measured on Pn or Sn on the WA simulated trace, dist is in km, S and E are the station and event (or source region) correction terms. For the North Atlantic, $a = 1.86$ and $b = 1.62$.

The scale is activated with RESET TEST(117).

The scale is defined in mbn.par, an example based on [Kim and Ottemöller, 2017] is:
# definition of scale for Pn and Sn
# MN = \log_{10}(A) - a \log_{10}(\text{ref/distance}) + S + E + b
#
#    a    ref    b
SCALE MBPN +1.861 100. 1.618
SCALE MBSN +2.176 100. 1.426
#
# group velocity windows for Pn and Sn
#
GROUP VEL PN 5.0 9.0
GROUP VEL SN 2.0 5.0
#
# station corrections (S)
#
<table>
<thead>
<tr>
<th>STATION CORRECTION</th>
<th>Pn</th>
<th>Sn</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARE0</td>
<td>0.168</td>
<td>0.168</td>
</tr>
<tr>
<td>BJ01</td>
<td>-0.217</td>
<td>-0.088</td>
</tr>
<tr>
<td>DAG</td>
<td>-0.285</td>
<td>-0.166</td>
</tr>
<tr>
<td>DBG</td>
<td>0.101</td>
<td>-0.052</td>
</tr>
<tr>
<td>HAMF</td>
<td>0.143</td>
<td>0.094</td>
</tr>
<tr>
<td>HOPEN</td>
<td>-0.348</td>
<td>-0.332</td>
</tr>
<tr>
<td>HSPB</td>
<td>0.047</td>
<td>0.086</td>
</tr>
<tr>
<td>JMIC</td>
<td>-0.308</td>
<td>0.143</td>
</tr>
<tr>
<td>KBS</td>
<td>-0.072</td>
<td>0.001</td>
</tr>
<tr>
<td>KEV</td>
<td>0.191</td>
<td>0.133</td>
</tr>
<tr>
<td>KTK1</td>
<td>0.072</td>
<td>0.155</td>
</tr>
<tr>
<td>LOF</td>
<td>0.153</td>
<td>-0.140</td>
</tr>
<tr>
<td>MOR8</td>
<td>0.065</td>
<td>-0.137</td>
</tr>
<tr>
<td>NOR</td>
<td>0.120</td>
<td>0.145</td>
</tr>
<tr>
<td>NSS</td>
<td>-0.046</td>
<td>-0.298</td>
</tr>
<tr>
<td>SCO</td>
<td>0.331</td>
<td>0.343</td>
</tr>
<tr>
<td>SPA0</td>
<td>-0.259</td>
<td>-0.086</td>
</tr>
<tr>
<td>STEI</td>
<td>0.023</td>
<td>0.002</td>
</tr>
<tr>
<td>TRO</td>
<td>0.120</td>
<td>0.028</td>
</tr>
</tbody>
</table>
#
# event corrections E
#
<table>
<thead>
<tr>
<th>name of area</th>
<th>minlon</th>
<th>maxlin</th>
<th>minlat</th>
<th>maxlat</th>
<th>Pn</th>
<th>Sn</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gakkel Ridge</td>
<td>-10.0</td>
<td>10.0</td>
<td>82.0</td>
<td>85.0</td>
<td>-0.090</td>
<td>-0.082</td>
</tr>
<tr>
<td>Spitsbergen TF</td>
<td>-10.0</td>
<td>10.0</td>
<td>79.7</td>
<td>82.0</td>
<td>-0.194</td>
<td>-0.352</td>
</tr>
<tr>
<td>Molloy FZ</td>
<td>0.0</td>
<td>10.0</td>
<td>77.0</td>
<td>79.7</td>
<td>-0.047</td>
<td>-0.118</td>
</tr>
<tr>
<td>Knipovich Ridge</td>
<td>0.0</td>
<td>11.0</td>
<td>73.0</td>
<td>77.0</td>
<td>+0.215</td>
<td>+0.215</td>
</tr>
<tr>
<td>Mohns Ridge</td>
<td>-4.2</td>
<td>8.0</td>
<td>71.0</td>
<td>73.0</td>
<td>+0.043</td>
<td>+0.277</td>
</tr>
<tr>
<td>Jan Mayen TF</td>
<td>-14.0</td>
<td>-4.2</td>
<td>70.0</td>
<td>72.0</td>
<td>-0.260</td>
<td>-0.077</td>
</tr>
<tr>
<td>Kolbeinsey Ridge</td>
<td>-20.0</td>
<td>-14.0</td>
<td>69.3</td>
<td>71.0</td>
<td>+0.274</td>
<td>+0.379</td>
</tr>
</tbody>
</table>

Coda magnitude $M_c$

The coda magnitude is calculated using
7.1. THE HYPOCENTER PROGRAM, HYP

\[ Mc = a \log_{10}(\text{coda}) + b \times \text{dist} + c \]

where coda is coda length in secs and a, b, and c are constants. If ‘a’ is given as a negative number, the following formula will be used

\[ Mc = \text{abs}(a) \times \log_{10}(\text{coda}) \times \log_{10}(\text{coda}) + b \times \text{dist} + c \]

If both \( Mc \) and \( Ml \) are calculated, \( Ml \) is written first on the header line.

**Coda magnitude is only calculated if the epicentral distance is less than TEST(57).**

Surface wave magnitude \( Ms \)

\[ Ms = \log_{10}(\text{amp}/T) + 1.66\log_{10}(\text{dist}) + 3.3 \]

where \( T \) is period. Amplitude is in micrometer and distance in degrees, however in the Nordic format nm and km are used and the program converts. \( Ms \) is only calculated if the period is larger than 10.0 seconds in which case the program automatically assumes that \( Ms \) is the wanted magnitude. The phase used can be AMS, AMPS, AMP or blank. The current version of MULPLT produces the standard IASPEI name IAMs,20. The many possible phase names are a result of changes over time and thus to ensure that \( Ms \) is calculated correctly with older data. It is assumed that the amplitude has been picked on a WWSSN standard LP trace and that the period is in the range 18 – 22s (see program MULPLT). \( Ms \) will be calculated even if the period is outside this range, but it will not be correct according to the standard. The distance range is between TEST(114) and 100 deg. before version 10.5, there were no distance limits. TEST(114)=20 deg by default. Depth must be less than TEST(115). Before version 10.5, there were no depth limit. TEST(115) is 60 km by default.

Broadband surface wave magnitude \( MS \) (IASPEI code MS,BB, but SEISAN uses MS for simplicity, new from SEISAN version 8.3)

\( MS \) is calculated using the standard

\[ MS = \log_{10}(\text{amp}/T)_{\text{max}} + 1.66\log_{10}(\text{dist}) + 3.3 \]

or

\[ MS = \log_{10}(V_{\text{max}}/2\pi) + 1.66\log_{10}(\text{dist}) + 3.3 \]

where \( V_{\text{max}} \) is the maximum velocity. The IASPEI definition is to use velocity and the period is thus not needed but read for information. The velocity is in micrometer/s and distance in degrees, however in the Nordic format nm/s and km are used and the program converts when calculating magnitudes. \( MS \) is only calculated if the period is larger than 3 seconds and less then 60 seconds, distance must be larger than or equal to 222 km (2 degrees) and less or equal to 160 degrees. The depth must be less than TEST(115) (default 60 km). Before version 10.5, there was no check of depth. The phase used to report the amplitude and period must be called IVMs,BB which the current version of MULPLT produces. The biggest advantage using \( MS \) compared to \( Ms \), is that any period in the range 2 – 60s can be used.

Body wave magnitude \( mb \)
mb is calculated using

\[ mb = \log_{10}(amp/T) + Q(dist,depth) \]

where Q is a hardwired function of distance and depth and amp is the amplitude in nm. There are two possibilities: The default (set by REST TEST(108)) is the standard Gutenberg and Richter (1956) curve while alternatively the Veith-Clawson curve can be used \cite{Veith and Clawson, 1972}. Before SEISAN version 8.3, Veith-Clawson was always used. mb is only calculated if the epicentral distance is less than or equal to 100 degrees and larger than or equal to TEST(113), default 20 deg (IASPEI standard and SEISAN default is 20 degrees). Before version 10.5, the lower distance was TEST(57) (default 1500 km). The period must be smaller than 3s and larger then 0.2 s and the phase is P, AMP, AMb, AMB, AMPb, blank character or IAMb. The current version of MULPLT produces the standard IASPEI name IAMb. The many possible phase names are a result of changes over time and thus to ensure that mb is calculated correctly with older data.

Broad band body wave magnitude mB (new from SEISAN version 8.3)

The broad band magnitude mB (official IASPEI name is mB_{BB}) is calculated using

\[ mB = \log_{10}(amp/T)_{\text{max}} + Q(dist,depth) \]

or

\[ mB = \log_{10}(V_{\text{max}}/2\pi) + Q(dist,depth) \]

where \( V_{\text{max}} \) is the maximum velocity and Q is a hardwired function of distance and depth. The IASPEI standard is to use velocity and SEISAN store the velocity in nm/s. There are two possibilities for the attenuation function: The default (set by RESET TEST(108)) is the standard Gutenberg and Richter (1956) curve while alternatively the Veith-Clawson curve can be used \cite{Veith and Clawson, 1972}. mB is only calculated if the epicentral distance is less than or equal to 100 degrees and larger than or equal to TEST(113) (IASPEI standard and SEISAN default 2 degrees. Before version 10.5, the lower limit was TEST(57), default 1500 km) and the period is larger than 0.2s and less than 30s and the phase name is IVmB_{BB}. The current version of MULPLT produces the standard IASPEI name IVmB_{BB}. The biggest advantage using mB compared to mb, is that the mB scale does not saturate before magnitude 8.

Moment magnitude Mw

Mw is calculated as

\[ Mw = \frac{2}{3} * \left( \log_{10}(\text{moment}) - 9.1 \right) \]

where moment is in Nm (see also section 8.9). When an event is relocated, the moment is also recalculated according to revised hypocentral distance. Mw is only calculated from moment as given on the SPEC lines, see MULPLT.

NOTE: If an amplitude has a given period between 5 and 10 secs, it is not used for Ml and mb magnitude calculation, see above. If an event is not located, there will normally be no magnitude calculation and all magnitude and distance information is deleted from the output S-file (hyp.out) except, the magnitude in the 3rd position on the header line if it has an agency different from the default agency. The only exception is that if a coda is given, the epicentral distance is retained and coda magnitude will therefore
be calculated. This means that for events, which cannot be located, it is still possible to calculate coda magnitudes by manually entering the epicentral distance on the line containing the coda length.

On the first header line, there is room for 3 magnitudes. If there is a magnitude in the 3rd position, it is not overwritten unless the default agency is overwritten, so there will often only be room for 2 calculated magnitudes on the first header line. If more magnitudes are calculated, they will be written on a subsequent hypocenter line, which is identified by having the same year, month, day and hypocenter agency as the first header line. This means that there is room for a total of 6 magnitudes, which can each, be updated when relocating. Hypocenter info and all 6 magnitudes can be printed out on one line with program REPORT.

All magnitudes can have a station dependent correction given in the station file. This correction does not affect the Mc in print.out file. Mb and mB use the same correction and Ms and MS use the same correction. Only calculate magnitude: If TEST(106) is set to 1.0, only magnitudes are calculated, provided a distance is given.

7.1.4 Use of S-P and L-S differences

Uncertainty in absolute times often makes it necessary to be able to use the difference in time between two arrivals such as P and S or P and Lg. If no absolute times are available, the calculated origin time will be close to that at the first arrival station and is of course meaningless. However, a perfectly good epicenter and depth can still be obtained from P-S or P-Lg differences alone. To enable this feature, set the weight for the P phase input record to 9. This P is then assigned a weight of 0, effectively disabling its use. However, a time residual and azimuth, etc., will still be calculated for it, enabling an assessment to be made of its absolute time. A search will then be made of the entire input phase set for an S or Lg phase at the same station. If such a phase is found, its variables are used to store the observed and calculated difference times and their derivatives, and it’s weight (0-4) is used for the difference phase. DON’T SET IT TO 9!! If two or more such phases (e.g., SN, SG, LG, etc.) are found, all their differences with the P time will be used instead of their absolute times. Blanks will appear beneath ‘hrmn’ in the residual summary for all such phases, while the observed and calculated difference times with the first P will appear beneath ‘t-obs’ and ‘t-cal’.

NOTE: Difference with Rg has not been implemented, could be done if need be.

NB. There must be at least one phase with absolute time to get a location.

7.1.5 Global event location

When locating globally, the program uses the IASPEI91 travel time software described by Buland and Chapman (1983) and Kennett and Engdahl (1991). The global model is used if the distance indicator in the S-file is D. If the distance indicator is R, the local model is used unless there are distant stations (distance larger then TEST(57) in which case the global model is used.

HYP evaluates all the IASPEI91 phases (up to 60) at each delta, and searches for the phase specified in the 4-character phase identifier. If no phase is found, the phase is given a weight of -1, which effectively removes it from the phase set. If a phase is labeled as ’P’, ’S’, ’PKP’ or ’SKS’, and this phase is not in the IASPEI91 list, the first arrival phase having P or S as its first letter is used, or PKP, SKS as its first 3 letters. In addition, include the PKiK phases in this search for ’PKP ’ and ’SKiK’ phases in the search for ’SKP ’. The IASPEI91 phase set currently includes: P, Pdiff, PKP, PKiKP, p, pPdiff, pPKP, pPKiKP, Sp, sPdiff, sPKP, sPKiKP, PP, P’P’, S, Sdiff, SKS, sP, sPdiff, sSKS, Ss, sSdiff, sSKS, SS, S’S’, PS, PKS, SP, SKP, SKiKP, PsP, PsS, ScP, ScS, PKKP, PKKS, SKKP, and SKKS.
Long phase names:
Normally SEISAN and the Nordic format assume up to 4 character phase names. However, when working with global phases, the phase name length can in a few cases be up to the ISC standard of 8 characters. The program then uses column 9 for weight (normally blank) and column 11-18 for the phase. In this case it is not possible to give a polarity.

In the new Nordic format, there is always room for 8 characters and a polarity can be stored.

old:

```
STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO AIN AR TRES W DIS CAZ7
KT1 S23 sPKiKP 654 2.96
```

new:

```
STAT COM NTLO IPHASE W HHMM SS.SSS PAR1 PAR2 AGA OPE AIN RES W DIS CAZ7
KT1 S Z sPKiKP 3 654 2.960 D
```

7.1.6 Criteria for a solution and weighting

The cases where a solution will not be attempted are as follows:

1. Multiple phases at two stations, but no azimuths. This is a non-unique case, even though four different arrivals are present.

2. Less than three phases from three different stations and no azimuths.

3. A single phase at one station with an azimuth.

Note that if phases are weighted out due to large distance or a bad fit during the first iteration, there might not be a location even if more than 3 stations are available.

Weighting:
A number of different weights may be used to calculate the solution.

1. User specified weights: These are calculated using the HYPO71 style weight number 0 to 4, read with each phase, where 0 corresponds to $w_1=1.0$, 1 to $w_1=0.75$, 2 to $w_1=0.5$, 3 to $w_1=0.25$ and 4 to $w_1=0$. Uncertain time is 9 meaning that absolute time is not used, see also use of S-P times on previous page.

2. Distance weighting: This is given by the formula $w_2=(x_{far}-\delta)(x_{far}-x_{near})$ where $\delta$ is the distance (km) of the event from the station and $x_{near}$ and $x_{far}$ are read from the station file, `STATION0.HYP`.

7.1. THE HYPOCENTER PROGRAM, HYP

4 Azimuth weighting: Azimuth residuals are divided by test(52), which is the error in azimuth that corresponds to a one-second error in arrival time. For example, if test(52)=5 (default), a phase residual of 5 degrees will become a residual of 1 (5/test(52)) in the parameter corrections and rms calculation.

All the above weights are multiplied together to calculate the weight used in the inversion. If the user-specified weight, w1, is changed by (2) or (3) above, changed to zero by the consistency check, or set to -1 because the phase is not recognized, an asterisk will appear after the final weight in the residual printout.

7.1.7 Eliminating outliers

Outliers can be hard to spot since the residuals from one or several outliers tend to be distributed over all residuals. The problem is particularly large when dealing with automatic picks (A in column 16 (old format) or 26 i (new format) of phase line) where gross mistakes, like picking P at S, can occur. One possibility is to use residual weighting but often more drastic means are needed. Fernando Carrilho (fernando.carrilho@ipma.pt) has made an algorithm for HYP which effectively finds and remove and/or weight out large outliers. Only P and S-phases will be checked. The procedure works for both local and distant events although it is mainly meant to be used with local events. The principle is the following:

Run the location in a loop (iterations) of a maximum of 15 steps

For each step:
- Weight out residuals larger than 99
- S-phase: If residual larger than 1.5*rms and larger than RESET TEST(109) it is weighted out. If in a following iteration it is smaller than 1.5*rms, and smaller then RESET TEST(109), it is included again. If 1.5*rms is less than 0.6 and the residual is less than 1.0, the data is always used. Default value of RESET TEST(109) is 0.8.
- P-phase: Same as for the S-phase but only after the 2. iteration.

The loop will be terminated if any of 2 conditions are met:
- rms is less than 0.3 and iteration is more then 5.
- Change in rms from one step to the next is less than 0.05 and iteration is larger than 4.

The phases with high residuals are now weighted out. They can also be removed from the S-file if the residuals are larger than a user defined value given in RESET TEST(110) (default value of 3 ), however this is only possible for the phases flagged as automatic.

There is also an option for keeping only the phase with the lowest residual if the same phase has been picked on several components of the same station. Phases will be deleted for both manual and automatic phase picks. Set RESET TEST(111)=1.0. Default value is 1.0.

There is an option for testing for Sg phases. If RESET TEST(112) is set to 1.0, there will be an additional iteration loop, where all S- phases which are only labeled S have been changed to Sg. At the end, the data set with the highest number of accepted phases will be used and all S's will be S or Sg. The phases which already have two letters like SG or SN will not be changed. This option will not be used for distant or regional events.

The default values are the ones best fitting automatic operation using AUTOPIC in the Portuguese network. Increasing the values of test(109) and test(110) will increase the number of accepted phases.
7.1.8 Determining which travel time software is used

The parameter test(57) is used to determine whether a layered model or IASPEI91 software is used to calculate the travel times and their derivatives. For the initial starting location, the distances from each station are calculated and IASPEI91 is used if any of them exceed test(57). However, this can be overridden by the distance indicator in column 22 of the Nordic header record. If this is L, a crustal model is used regardless of distance, whereas if it is D, IASPEI91 is used, while R has no effect i.e., test(57) is still used. So if either a crustal model or IASPEI91 tables are wanted, use either L or D respectively.

7.1.9 Starting location

The program uses a starting location algorithm (reset test(56)) which tests the rms of all starting locations and select the minimum rms solution, see HYP manual.

User defined start location: If an S is written in the input S-file at column 45 of the epicenter line, the location starts at the location (epicenter) given on the header line. If an S is written in column 44 on header line, the depth iteration will start at depth given on the header line. If N is written in column 45, the nearest station will be used irrespective of global settings.

Starting depth:
If no event specific start depth is given in S-file, the starting depth is taken from the first number on the control line (see later) in the HYPO71 style. However, there is often problems obtaining a reliable depth due to local minima. This can be manually checked with program RMSDEP from EEV. HYP can also be set up to locate the same event starting with a range of different start depths, and then choose the one with the lowest RMS. This can significantly improve the reliability of depth determination. Selecting 3 to 5 different start depth is often enough. This option is set on the control line in the station file.

Fixing location:
Using F instead of S, fixes the position (depth and location).

Do not locate event:
If a * is written in column 45, the event is not located, can be used if an external location is to be kept unchanged.

Only calculate magnitudes and update spectral values
Set TEST(106) to 1.0

Fixing origin time:
Using an F in column 11 of header line will fix the origin time given on the header line.

If both depth and location are fixed, but not the origin time, new origin time and residuals will be calculated. This can be useful when working with readings from a few stations which should be checked against known locations. If e.g. distant events are read, it is often the practice to put in the PDE location on the header line and calculate residuals relative to the observations. When the UPDATE is made, the agency of the location is NOT changed, assuming that if both depth and epicenter are fixed, the hypocenter must come from an external agency.
7.1. Preventing a local minimum

For some data, the iterations will stop in a local minimum. This can partly be checked with the DRMS output in the print.out file. The DRMS output gives the RMS in 6 points around the hypocenter at distance DRMS. DRMS is set with RESET TEST (13). In order to check for a local minimum, a more extensive grids search can be done. This is done by using RESET TEST 119-121. Setting these parameters, a grid search can be done in 3D around the hypocenter in a user selected grid size. The grid search is done after the last iteration. If a point is found with a lower RMS than given by the iterative process, a new iterative process is started with that point as start location. The grid search can be particularly useful for local earthquakes, however, in the current implementation it is also done for distant earthquakes using the same parameters. Note that using a large grid might significantly slow down the processing time.

When using the grid search option for start location, an extra ouput is given in the print.out file:

Grid rms smaller than solution rms 1.953 3.391
Hypocentral dist. (km) from epicenter to grid point min. 28.3
Depth change 10.00
Iterate again with new start location: 62.303 5.760 15.0

This information gives how far away the minimum is and what the RMS is.

7.1.11 Alternative model

By default, an event is located using the \texttt{STATION0.HYP} input file. However, each event can use its own model (with all the location parameters) which is specified with one character in column 21 on the Nordic input file header line. The model then has a corresponding name. If e.g. the model is called W, the corresponding input station file will be called \texttt{STATIONW.HYP}. It is therefore possible to have as many different station files, as there are printable characters. Note that if a different model x has been specified and is not present, the program will stop with the message "\texttt{STATIONx.HYP does not exist}". The file \texttt{MODEL.DEF} in DAT can be used to assign the single character a name, which can be listed from EEV. The format in \texttt{MODEL.DEF} is one line per model, the model indicator is given in column 1, column 2 is blank and the model name is given in columns 3 to 80. The \texttt{MODEL.DEF} is for information only.

7.1.12 Location by grid search

Hypocenter can also locate an event by grid search. This can be quite time consuming but illustrates well the problem with local minima. Grid search can only be done for one event at a time using EEV command 'grid'.

Hypocenter is started up and will ask for the grid: Latitude and longitude range and grid spacing. A maximum of 71 points can be used in each direction. The point with the lowest RMS is found and the corresponding location and residual is printed on the screen. It is now optionally possible to plot the contours on the screen. The map coordinates used are as defined in SEISAN.DEF. Note that the grid search is using exactly the same parameters as Hypocenter. This includes all weights and phase types. The depth is fixed to the depth given in the S-file header line. For more details and an example, see application note epi.pdf in INF.
7.1.13 Using HYP to determine crustal structure

HYP has an option to locate a data set for a large number of different models and then determine which model gives the lowest average RMS for the data set. This might be a useful option, particularly when a sparse data set is available. In order to use this option, an additional input parameter file `h_models.par` is given. When this file is in the working directory, HYP will switch to multiple model mode. SO ONLY HAVE THIS FILE IN WORKING DIRECTORY IF MULTIPLE MODEL MODE IS INTENDED. When using this option, all events must use the same `STATIONx.HYP` file, otherwise the program fails. The input MUST be from a single file, NOT from the data base. THE PROGRAM MUST RUN IN NON INTERACTIVE MODE. Below is an example of an input file.

```
layer # start vp delta vp # delta start h delta h # delta
1   4.55  0.1    5   0.0    1.0    1
2   6.3   0.1    5   4.0    1.0    1
3   6.8   0.1    5   22.5   1.0    1
4   7.90  0.05   3   32.5   1.0    1
4   8.05  0.05   4   40.0   1.0    1
```

The first line is info only. Layer # is also only for information. For each layer, there is a start P-velocity (start vp), increment in velocity (delta vp) and number of increments (# delta). The following inputs are then the same for layer depths. There must be an entry for each layer even if no variation is used. In the above example, no variation in layer thickness is tested for. An example input file is given in DAT. The parameters for location not set in `h_models.par` like Vp/Vs, Lg velocity etc remain unchanged. When HYP starts up, it will print out how many permutations are required. If more than a few thousand, reduce the number of models. In any case it is an advantage to first try with just a few models to get a feeling for how sensitive the data is for model changes.

An output file `h_models.out` is generated, see example below. For each model tested, one output line is given with the RMS and the model. In the example below only the last 5 models are shown. Since many models can have very similar average RMS, the best 10 models are printed at the end.

```
0.946  4.95  0.00  6.70  4.00  7.20  24.50  7.90  32.00  8.20  40.00
2.607  4.95  0.00  6.70  4.00  7.20  24.50  7.90  32.00  8.30  40.00
0.934  4.95  0.00  6.70  4.00  7.20  24.50  8.00  32.00  8.10  40.00
0.994  4.95  0.00  6.70  4.00  7.20  24.50  8.00  32.00  8.20  40.00
2.677  4.95  0.00  6.70  4.00  7.20  24.50  8.00  32.00  8.30  40.00

Minimum rms  0.764057

The best models
```

```
0.771  4.95  0.00  6.50  4.00  7.10  24.50  7.80  32.00  8.10  40.00
0.766  4.85  0.00  6.50  4.00  7.00  23.50  7.80  32.00  8.10  40.00
0.767  4.85  0.00  6.50  4.00  7.00  24.50  7.80  32.00  8.10  40.00
0.769  4.85  0.00  6.50  4.00  7.10  23.50  7.80  32.00  8.10  40.00
0.766  4.85  0.00  6.50  4.00  7.10  24.50  7.80  32.00  8.10  40.00
0.772  4.85  0.00  6.50  4.00  7.20  24.50  7.80  32.00  8.10  40.00
0.771  4.95  0.00  6.50  4.00  6.90  22.50  7.80  32.00  8.10  40.00
0.771  4.95  0.00  6.50  4.00  7.00  22.50  7.80  32.00  8.10  40.00
0.770  4.95  0.00  6.50  4.00  7.00  23.50  7.80  32.00  8.10  40.00
```
7.1.14 Running HYP

The program is started with command HYP from the prompt line (interactive mode) or with ‘L’ in EEV or SE. HYP can also be started with an argument like hyp input.dat, where input.dat is an S-file. The first event in the S-file will then be located without further user interaction. Other prompt options (mainly used in connection with EEV) are

-seisanexp Use with SE
-update Stop at end of run, ask if update and overwrite input file
-op XXX Operator XXX to use in -update option chosen

Below follows an example of running outside EEV, explanations are in lower case. Note that the STATION0.HYP file MUST be present in the DAT directory for HYP to know that it is working with a SEISAN database. If not present, HYP will only ask for an input file name, see HYP manual.
CHAPTER 7. HYPOCENTER LOCATION PROGRAMS: HYPOCENTER, HYPO71 AND HYPoinverse

HYP

Arrival time data input, select one:

- SEISAN database or : RETURN
- Alternative database, give 1-5 letter code :
- Local index file, name must start with index or :
- Local database, write ,, or :
- File name for one file in NORDIC format :

Your answer here determines the input source. A return means that you work directly on the BER database. A 1-5 letter code gives name of database, e.g. NAO. An index file or the name of a readings file is used when you want to work on specific subsets.

Local database is S-files in local directory.

Start Time (YYYYMMDDHHMMSS) : 199012
End Time, RETURN is to end of month : 19901205
Standard formatted time input.

Interactive operation (N/Y=return)

If N, whole time interval or file is located, one line output per event.

# 1 1992 12 3 0137 40.3 NPHS= 12 T Q L #XXX
# 2 1992 12 3 0237 43.3 NPHS= 14 T Q L #XXX l ! now locate
here comes location, see HYP manual*******************************
# 2 1992 12 3 0237 43.3 NPHS= 14 T Q L #XXX q ! stop

PRINT OUTPUT IN FILE print.out
CAT-FILE IN FILE hyp.out
Summary file in hypsum.out

In interactive mode, as shown above, event date is printed out for each event and action is taken as in EEV for the options available. If HYP run on a single file, the options above are available meaning that HYP can select and locate different events in a single file using the event number. If HYP runs on a database, the EEV options D and B are also available, but not shown. If the option of no interactive input is chosen, the program will locate from beginning to end without any more user interaction. This is a useful option for testing a subset of the database with different models etc. without changing the database. Note that the input file or database is never overwritten by HYP.

ALL TYPE ONE LINES WITH SAME AGENCY AS GIVEN IN STATIONX.HYP FILE WILL BE DELETED SO THERE WILL NEVER BE MORE THAN ONE TYPE 1 LINE IN OUTPUT WITH CURRENT AGENCY (except possibly a second magnitude line with a different type magnitude as given on main header line).

Problems: Sometimes HYP will not locate an event, look in the print.out file to see what happened. In EEV this can be done with command ep. In some cases, the initial location was put beyond the limits set by the parameters. If e.g. an event is defined as a local event and no readings are to be used further away than 2000 km (distance weighting, see following table or TEST(41)) then no location will
be attempted. Try to change the event type to D and see if the event locates. In a few other cases it might be an advantage to use a starting location.

7.1.15 Station and model files

Station input is given in near standard HYPO71 format in the file STATION0.HYP in directory DAT. If however the user wants to try a different model without changing the standard model in DAT, this is possible by having a STATION0.HYP file in the working directory, since the program always looks there first for the STATION0.HYP file (see example at end of this section). Another possibility is to use another model for just one event by setting a flag in the phase input file, see below.

The station file included with SEISAN usually includes the complete ISC station list. The ISC station list can be downloaded from ISC and converted to SEISAN format with program ISCSTA in order to make updates. ISCSTA can also select out of the large station file (in ISC for SEISAN format) all stations corresponding to events in a CAT file so it is possible to make a much smaller station file for the users data.

Below is an example of a STATION0.HYP file. The format is close to the HYPO71 format with one extra line at the bottom. In addition, 5 letter station codes can be used. The test parameters 2-13 are as in HYPO71, see also HYPOCENTER manual section 4.1.2.

Comments are given after !’s
RESET TEST(01)=0.3  
RESET TEST(03)=0.6  
RESET TEST(06)=0.1  
RESET TEST(07)= 3.0  
RESET TEST(08)=2.6  
RESET TEST(09)=0.001  
RESET TEST(11)=50.0  
RESET TEST(13)=5.0  
RESET TEST(50)=1.0

! one and only one blank line here

UPP 5951.50N 1737.60E 14 ! station lines
CDP 5541.00N 1226.00E 13
KBS 7855.08N 1155.44E 46
EBH 5614890N 330490W 375 ! high accuracy lat-lon
OSG 6029.80N 252.55E-100  
01A06049.43N 1049.95E 426
BERGE6057.12N 1133.15E 100 ! 5 char station name
-BEBGE6157.12N 1133.15E1100 ! 5 char station name and at 1100 m depth

...  

! one and only one blank line here

6.2 0.0 ! ! model lines
6.6 12.0
7.1 23.0 3.8 2.2 200.0 300.0 **
8.05 31.0 N ! N indicates location of Moho
8.25 50.0
8.5 80.0 !
15. 600. 1300. 1.73 5 5.0 10.0 ! control parameters
BER ! Reporting agency (a3)

Format of the station line is 2x,a4,i2,f5.3,a1,i3,f5.3,a1,i4,f6.2,5f5.2,9f6.2,2a2 or 1x,a5 .... if the station has 5 characters. The content is:

station code 4-5 chars (see above)
latitude in degrees
latitude in min
north or south (N or S)
longitude in degrees
longitude in minutes
east or west (E or W)
altitude in m, in some rare cases, the station is deeper than 1000 m in which case the minus sign has to be put in column 1
P-delay in secs, S-delay is the same multiplied by Vp/Vs as given below. Since it is a delay, it is subtracted from the arrival time.
Magnitude corrections for the magnitudes: Mc, MI, mb or mB, Msi or MS and Mw
Spherical harmonic station corrections Network and Location codes

NOTE: Network and location are new from version 12 and are only used with new Nordic format. The are located in column 113-116. If the station file has network and location and the S-files does not have
it, the station will be selected anyway. So network and location is really only needed if two stations have
the the same station code but different location or network.

NOTE: The format for the minutes is f5.3. Normally minutes are given as e.g. 20.22, however if a higher
accuracy is needed, it can be written without the point as 20222 meaning 20.222 according to the format
f5.3. See also section 5.3.5.

The command GET_STAT (get_stat station-code;) can be used to check that a station is present in the
station file and that the coordinates are correct. An example is

c:\seismo\WOR> get_stat BER
BER  60.3838  5.3338  50

The magnitude residuals are added to magnitudes calculated for each station but the result is only seen
in the final average magnitude. If the magnitude correction is set to 99.0, the magnitude is not used in
the average. The magnitude corrections for mb and mB are the same and similarly also for Ms and MS.

Format of model line: 3f7.3, a1, 3f7.2. The information is:
P- velocity (km/sec)
Depth to interface (km)
S- velocity (not needed). If no S-velocity is given, the S-velocity is calculated from the P-velocity using
the given Vp/Vs ratio.
Interface indicator: N: Moho, B: Conrad
Density (g/cm**3) (not needed)
Qp (not needed)
Qs (not needed)

Density and Q is only used by modeling programs and moment tensor inversion. In this way the station
file is a complete model file for making synthetic seismograms. NB: Moho cannot be the last layer, there
MUST be one layer below interface marked with N.

The line with ** indicates optional Vs, density, Qp and Qs. This is information only used with modeling,
see section 29. Format for additional info is 25x, 4f10.1.

Format of control line: 3f5.0, f5.2, i5, 2f5.1. The information is:
start depth in km, used if no range of start depths specified (see below)
xnear: distance at which distance weighting start
xfar: distance at which distance weighting is zero, beyond xfar, the phase is not used (local events only)
Vp/Vs ratio
number of start depths
start depth of range of start depths
increment in start depths
NB: If these parameters are used, the fixed initial start depth is not used

The input at the bottom is reporting agency used for both hypocenter and magnitudes.

Since the program locates distant events, max distance, reset test(41) must be set to a large value. To
avoid that local events move out in the blue, the parameters xnear and xfar must be set not larger than
2000 to 3000 km. Xnear and xfar are only used for local events (flag L) and regional events if the local
crustal model is used.
7.1.16 **RESET TEST parameters**

HYP will assign reasonable default values for RESET TEST parameter. Below is shown a summary. For full details see HYP manual. The number to the left is the control parameter and D indicates the default value. The most important parameter are given in bold.

2: Step length damping control, D: 500.0.
7-9: Duration magnitude coefficients used for calculating the coda magnitude, as \( \text{MAG} = \text{TEST}(7) + \text{TEST}(8) \times \log(T) + \text{TEST}(9) \times \text{DELTA} \) where \( T \) is the coda length in seconds, \( \text{DELTA} \) is the hypocentral distance in km. D: 7: -0.87, 8: 2.0, 9: 0.0035. [Lee et al., 1972] If test(8) is negative, its positive value will be used and log(T) will be squared. Note however, that the individual stations magnitude values printed out during the run of HYP still will be using the unsquared log(T).

11: Maximum no of iterations in the least-squares rms minimization, D: 99.0

13: Increment in km for auxiliary rms, D: 20.0 km. To disable (save some computation time), set to 0.0.

30: Initial damping factor, D: 0.005

31: Max degs of freedom: Set to 3 for determining origin time and hypocenter, set to 2 for fixed depth solution (depth on phase headers), -2 fix all events to starting depth in STATION0.HYP, 1 to fix all hypocenters to value on phase headers, 0 to fix hypocenters and origin times to values on phase headers. D: 3.0

32: Magnitude of parameter changes (km) below which convergence is assumed, D: 0.05

34: Minimum spread to normalize residuals, D: 0.1, do not change

35: Bisquare weighting width, D: 4.685, do not change

36: RMS residual low limit for bisquare weighting for local events, D: 0.0

37: Maximum number of increases in damping before fixing depth, D: 10.0

38: Least squares errors (0.0), damped least squares errors (1.0) with initial test(30) damping value, D: 0.0

39: Factor by which damping is increased when RMS increases, D: 4.0

40: Depth origin of coordinate system, 0: sea level, 1:maximum elevation station in station list, D: 0.0

41: Maximum distance (km) from nearest station at which hypocentral solutions will be generated, D: 20000.

43: Maximum rms for an event to be used in average station residual calculation - doesn’t affect the final hypocenter solution, D: 1.5

44: Rg phase velocity in km/sec, D: 3.0

45: Minimum rms difference between the location on the header line and the new location for the event to be used for average difference in location, D: 50.0

46: Minimum number of non zero weight phases for event ot be included in average difference in location, D: 3.0

47: Prevent depth to go below Moho and Conrad for n and b phases respectively, 1: enabled, 0: disabled, D: 0.0

49: T-phase velocity, D: 1.48 km/sec

50: Flag for using azimuth phases, 0 disables. Disabling the azimuths also means that they are not used for a starting location. A better solution will often be to set the azimuth error, TEST(52) to a large value, effectively disabling them.D: 1.0 (enabled).

51: Lg phase velocity in km/sec, D: 3.5.

52: Relative weighting of error in azimuth used in azimuth inversion (degrees). The default value of 10 means that an error of 10 degrees will give the same contribution to the rms residual as a travel time error of 1 sec, D: 5.0
7.1. THE HYPOCENTER PROGRAM, HYP

53: Critical distance phases moved to by start loc. if Pn or Sn, D: 130.0 km
56: A value of 1.0 enables the starting location algorithm, STARTLOC. Estimates are then obtained from apparent velocity, distance, azimuths, etc. If test(56)=0.0 epicenter is taken 0.2 km from the first arrival station. D: 1.0 MUST BE SET TO 1.0 TO LOCATE WITH ONE STATION ONLY.
57: Distance (geocentric km) beyond which IASPEI91 tables are used to calculate travel times for regional events. Can be overridden by the distance letter L in the Nordic format. D: 1500 km
58: Maximum apparent velocity (km/sec) for phase data to be used. This option was added to selectively disable some of the PKP phases, which have large errors due to their steep angle of incidence. Their velocities were almost always \( \approx 25 \text{ km/s} \), D: 100.0 (effectively disabled)
59: Critical distance for PKP core phases to be used in starting location, D: 13000 km
60: Seconds by which the arrival time difference between two adjacent stations can exceed the travel time between them. Setting this to 0 disables the initial consistency check. D: 5.0
61: Multiple of apparent velocity regression residual rms at which arrival times are weighted to zero during start location determination. Reducing this value will cause arrivals to be rejected when they do not conform to the plane wave set of arrivals which is characteristic of distant events. Unless you are getting a lot of messages 'xxx removed: Apparent velocity deviation =..', in the output, it is recommend against changing this default value. However, you can disable this feature by setting test(61)=0.0, D: 2.0
62: Use of IASP91 phases.0: Only calculate ‘basic’ phases, 1: calculate all, D: 1.0
63: Types of phases used when calculating travel time, D: 0.0
64: Allow temporary increase in RMS by this factor, D: 2.0
65: Number of iterations for which increased rms is allowed, D: 3.0
66: Print out of travel time calculation errors (1=y,0=n), D: 0.0
67: Recognize blank phases as P (y=1,n=0), D: 0.0
68: Apparent P-velocity(km/sec) to calculate start depth from pP-P, D: 5.0
69: Distance (deg) beyond which PKiKP or PKP is used as first arrival instead of Pdif D: 110.0
70: Maximum depth that the hypocenter is allowed to move to, D: 700 km
71: Sort output according to distance,(y=1,n=0), D: 1.0
72: Auto phase identification for distant events (y=1,n=0), D: 0.0
73: Number of iterations with first P’s before autophase id., D: 3.0
74: Print input phase data in print.out (y=1,n=0), 0.0
75-78 Ml magnitude coefficients. Ml = TEST(75)*log10(amp) + TEST(76)*log10(dist) + TEST(77)*dist + TEST(78) where amp is amplitude in nm and dist hypocentral distance in km. The defaults are Ml = 1.0 * log10(amp) + 1.11*log10(dist) + 0.00189*dist - 2.09 which is close to the original Richter definition [Hutton and Boore, 1987].
79: Minimum number of stations to attempt a solution,D: 1.0
80: Minimum number of phases (azimuth is counted as a phase) to attempt a solution, D: 3.0
81: Disable location of local events if 0.0, D: 1.0
82: Disable location of regional events if 0.0, D: 1.0
83: Disable location of distant events if 0.0, D: 1.0
84: Disable ellipticity correction for distant events if 0.0, D: 1.0
85: A priori error(sec) of local events. This affects the error estimates, particularly when few stations are present. D: 0.1. See TEST(91) for distant earthquakes.
86: Number of degrees of freedom in estimating test(85) for loc. ev., D: 8.0
87: Confidence level that the solution will lie outside the confidence ellipse defined by the covariance matrix. The default value corresponds to 90 % confidence., D: 0.1
88: RMS residual(sec) at which residual weighting is applied for distant events. Set to 0.0 to disable. D: 10000.0
89: Use depth phases (y=1,n=0), D: 1.0
90: Use of core phases (y=1,n=0), D: 1.0
91: Same as TEST(85) for distant events, D: 1.0
92: Number of degrees of freedom for test(91), D: 8.0
93: Output longitude to always be positive (y=1,n=0), D: 0.0
94: Value of residual below which zero weight phases (w=4) is used again, D: 0.0
95: Disable use of core phases between 135 and 150 deg, 1: disabled, 0: enabled, D: 0.0
96: Variation of depth to find minimum rms, 1: enabled, 0: disabled, D: 0.0
97: Minute error correction 1: enabled, 0: disabled, D: 0.0
98: Enable spherical harmonic station corrections, 1: enabled, 0: disabled, D: 0.0
99-101: Lg, Rg and T weights put in permanently: D: 1.0, 1.0, 0.0
99: Lg phase weight: multiplied by phase weight. D: 1.0
100: Rg phase weight: multiplied by phase weight. D: 1.0
101: T phase weight: multiplied by phase weight. D: 0.0
102: Not used
103: Minimum number of depth phases for starting depth, D: 1.0
104: Minimum distance of epicenter from array for distant events, D: 30.0 deg.
105: Enable gradient model, not yet implemented
106: Only calculate magnitudes and update spectral values, 1: enabled, 0: disabled, D: 0.0
107: Use xnear and xfar from sfile, 0: disabled (xnear and xfar from STATION0.HYP file), 1 enabled, D: 0.0 (see format description)
108: mb attenuation curve, 0.0 Richter, 1.0 Veith and Clawson, D: 0.0
109: In reject mode, if residual is larger than this, it is weighted out. D: 0.8
110: In reject mode, in final run, phases with residual larger this and automatic, are removed. D: 3.0
111: In reject mode, if several phases from same station, only keep the one with lowest residual if test is 1.0. D: 1.0
112: In reject mode, set this to 1.0 ito enable testing for Sg when rejecting phases, default 0.0.
113: Shortest distance for calculating mb and Mb. D: 2224 km (20 deg)
114: Shortest distance for calculating Ms. D: 2224 km (20 deg)
115: Maximum depth for calculating Ms and MS. D: 60.0 km
116: Average magnitude is simple average (0.0) or median (1.0) D: 0.0
117: Use mbn magnitude instead or ML if events are in defined areas D: 0.0
118: Decides if a local minimum should be searched for after iterations ends. If an RMS with lower rms than the iterated solution is found, new iterations will be done with that solution as a starting epicenter. If zero, grid search is not done, if positive number grid search is done with that number used as grid size (km). D: 0.0
119: The number of gridpoints in one quadrant so the grid will be +/- test(119) on each side. D: 10.0
120: Decides if depth grid search for a local minimum like test 118. test 118 and 120 can be combined. The value is the size of the depth step (km). D: 0.0 no grid search.
121: The number of steps in depth grid search. D: 10

The test parameter defaults are set in file hyposub1.for in LIB.
7.1.17  HYP output

Output from the program is a CAT-file (hyp.out) and the original HYPOCENTER print file (print.out) with more detailed information. The hyp.out file can be plotted directly using EPIMAP or MAP. In addition, there is also the HYPO71 style summary file, hypsum.out. NOTE: In print.out and hypsum.out, year is only given with 2 digits. Magnitude in hypsum.out and print.out are only coda magnitude and will be different from same magnitude in hyp.out if a magnitude correction has been used.

When HYP is executed from EEV, the print.out file has no station listing. In all other cases, there is a station listing.

Some explanation is given below, for details see HYP manual

The output in print.out first shows the content of the TEST parameters in the STATION0.HYP file. After that comes some routine output from the starting location algorithm. Then follows the output from the iterations, which should be self-explanatory. The location is then given on one line containing origin time, latitude longitude (deg min), depth, number of phases, the number of degrees of freedom in the spatial solution (maximum 3), rms damping and errors, error estimates, resolution matrix. Last are the station lines with the following abbreviations:

<table>
<thead>
<tr>
<th>stn</th>
<th>dist</th>
<th>azm</th>
<th>ain</th>
<th>w</th>
<th>phs</th>
<th>calcphs</th>
<th>hrmn</th>
<th>t-sec</th>
<th>t-obs</th>
<th>t-cal</th>
<th>res</th>
<th>wt</th>
<th>di</th>
</tr>
</thead>
<tbody>
<tr>
<td>stn</td>
<td>dist</td>
<td>azm</td>
<td>ain</td>
<td>w</td>
<td>phs</td>
<td>calcphs</td>
<td>hrmn</td>
<td>t-sec</td>
<td>t-obs</td>
<td>t-cal</td>
<td>res</td>
<td>wt</td>
<td>di</td>
</tr>
<tr>
<td>KIF</td>
<td>137</td>
<td>7.6</td>
<td>55.1</td>
<td>0 P</td>
<td>PN4</td>
<td>029.9</td>
<td>20.05</td>
<td>21.06</td>
<td>-1.01</td>
<td>1.00</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEF</td>
<td>152</td>
<td>63.2</td>
<td>55.1</td>
<td>0 P</td>
<td>PN4</td>
<td>029.3</td>
<td>22.61</td>
<td>22.93</td>
<td>-0.32</td>
<td>1.00</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KTK1</td>
<td>178</td>
<td>40.5</td>
<td>55.1</td>
<td>0 P</td>
<td>PN4</td>
<td>029.4</td>
<td>26.36</td>
<td>26.07</td>
<td>0.29</td>
<td>1.00</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STEI</td>
<td>215</td>
<td>75.5</td>
<td>55.1</td>
<td>0 P</td>
<td>PN4</td>
<td>029.0</td>
<td>30.50</td>
<td>30.66</td>
<td>-0.16</td>
<td>1.00</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FAUS</td>
<td>221</td>
<td>259.5</td>
<td>55.1</td>
<td>0 P</td>
<td>PN4</td>
<td>029.4</td>
<td>31.09</td>
<td>31.38</td>
<td>-0.30</td>
<td>1.00</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARA0</td>
<td>283</td>
<td>45.3</td>
<td>55.1</td>
<td>0 P</td>
<td>PN4</td>
<td>029.6</td>
<td>40.75</td>
<td>39.16</td>
<td>1.59</td>
<td>1.00</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An example of output on screen is given below:

```

<table>
<thead>
<tr>
<th>date</th>
<th>hrmn</th>
<th>sec</th>
<th>lat</th>
<th>long</th>
<th>depth</th>
<th>no m</th>
<th>rms</th>
<th>damp</th>
<th>erln</th>
<th>erlt</th>
<th>erdp</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>1</td>
<td>1</td>
<td>029</td>
<td>9.85</td>
<td>6749.40N</td>
<td>20.00</td>
<td>15.00</td>
<td>0.53</td>
<td>0.000</td>
<td>54.9</td>
<td>42.6118.6</td>
</tr>
<tr>
<td>KIF</td>
<td>137</td>
<td>7.6</td>
<td>55.1</td>
<td>0 P</td>
<td>PN4</td>
<td>029.9</td>
<td>20.05</td>
<td>21.06</td>
<td>-1.01</td>
<td>1.00</td>
<td>16</td>
</tr>
<tr>
<td>KIF</td>
<td>137</td>
<td>7.6</td>
<td>BAZ</td>
<td>7.6</td>
<td>PN4</td>
<td>029.9</td>
<td>218.0</td>
<td>22.93</td>
<td>-1.01</td>
<td>1.00</td>
<td>16</td>
</tr>
<tr>
<td>KIF</td>
<td>137</td>
<td>7.6</td>
<td>55.1</td>
<td>0 S</td>
<td>SN4</td>
<td>029.9</td>
<td>26.36</td>
<td>26.07</td>
<td>0.29</td>
<td>1.00</td>
<td>4</td>
</tr>
<tr>
<td>HEF</td>
<td>152</td>
<td>63.2</td>
<td>55.1</td>
<td>0 P</td>
<td>PN4</td>
<td>029.3</td>
<td>22.61</td>
<td>22.93</td>
<td>-0.32</td>
<td>1.00</td>
<td>12</td>
</tr>
<tr>
<td>KTK1</td>
<td>178</td>
<td>40.5</td>
<td>55.1</td>
<td>0 P</td>
<td>PN4</td>
<td>029.4</td>
<td>30.50</td>
<td>30.66</td>
<td>-0.16</td>
<td>1.00</td>
<td>10</td>
</tr>
<tr>
<td>STEI</td>
<td>215</td>
<td>75.5</td>
<td>55.1</td>
<td>0 P</td>
<td>PN4</td>
<td>029.0</td>
<td>30.50</td>
<td>30.66</td>
<td>-0.16</td>
<td>1.00</td>
<td>10</td>
</tr>
<tr>
<td>FAUS</td>
<td>221</td>
<td>259.5</td>
<td>55.1</td>
<td>0 P</td>
<td>PN4</td>
<td>029.4</td>
<td>31.09</td>
<td>31.38</td>
<td>-0.30</td>
<td>1.00</td>
<td>16</td>
</tr>
<tr>
<td>ARA0</td>
<td>283</td>
<td>45.3</td>
<td>55.1</td>
<td>0 P</td>
<td>PN4</td>
<td>029.6</td>
<td>40.75</td>
<td>39.16</td>
<td>1.59</td>
<td>1.00</td>
<td>4</td>
</tr>
</tbody>
</table>
```
In case of new format, it would be

```
16 1 1 029 9.60 67.4752N 20.2959E 15.0 3 0.59 0.000 45.2 20.4 95.9
KIF 140 6.1 55.1 0 P PN4 029 29.9 20.30 21.44 -1.14 1.00 29 F100
KIF 140 6.1 BAZ-P 169.2 186.4 -17.22 0.20 5
KIF 140 6.1 55.1 0 S SN4 029 46.4 36.80 37.30 -0.50 1.00 26
HEF 151 61.5 55.1 0 P PN4 029 32.5 22.86 22.74 0.11 1.00 9
KTK1 178 39.0 55.1 0 P PN4 029 36.2 26.61 26.13 0.48 1.00 4
STEI 219 276.4 55.1 0 P PN4 029 40.3 30.75 31.13 -0.39 1.00 10
FAUS 224 260.6 55.1 0 P PN4 029 40.9 31.34 31.73 -0.40 1.00 12
ARA0 283 44.3 55.1 0 P PN4 029 50.6 31.00 30.16 1.84 1.00 5
```

BAZ means there is a back azimuth for station KIF. For the new format, it is shown that the BAZ is on the P. In addition, KIF has NTLO shown at the end of the line. At the end of list is shown the original data (OLD) and the new hypocenter so it is easy to compare the effect of changing something. In the print.out file there are details of the difference between the two hypocenters.

A station weight wt=-1 means that the phase travel time could not be calculated. The output phases can be e.g. PN2, where 2 means that the phase calculated has been refracted in layer 2 and PN5 refracted in layer 5. The input phase is then just P and a local model is used.

Any change in the input phase ID is signified by an asterisk (*) before the phase ID.

If amplitudes are available, Ml, Mb, Mw or Ms will be calculated, and all stations calculating Ml, Mb, Mw or MS will additionally be displayed at the end of the interactive printout.

Change of day:

If the origin time of the located event occur on the day before the time in the header line, the time in the header line is changed to the previous day and all phase arrivals are changed accordingly. This means that some hour values will be more than 23 since phase arrival times refer to the main header.

Seismic moments etc: After locating an event, HYP will check if there is spectral information (Moment etc, see MULPLT) available in the S-file and average values will be calculated and written into the output file.
7.2. **HYPO71 (SUN ONLY)**

7.1.18 Problems

If no location or an obviously wrong location is obtained, check `print.out`. Common problems are:

- Wrong location: Program gets into a local minimum. Use the individual event start location option (`S` in header line). If the problem happen often, try either option for start location (nearest station or start location routine). If dept is the problem, try a range of start depths (set in `STATION0.HYP`).

- No location: The program iterates outside the maximum distance set for a local or regional event (RESET TEST 57) or the initial start location is outside limits. Use a fixed start location or check readings to get a better start location.

7.2 **HYPO71 (Sun only)**

By Brian Baptie, BGS

HYPO71 is a computer program for determining hypocenter, magnitude and first motion pattern of local earthquakes written by Lee et al. [1972] using a stepwise statistical regression procedure outlined in Draper and Smith [1966]. The user’s manuals were originally released by the authors as a series of open-file reports of the U.S. Geological Survey and contain a full description of input and output parameters and usage. The SEISAN version of the program is essentially the same as the original, the only differences being in the input and output facility. Input data required are phase arrival times, station co-ordinates and a crustal velocity model. SEISAN extracts the arrival information from a Nordic format phase readings file and the station and velocity information from the station input file `STATION0.HYP`, found either in the SEISAN data directory, DAT, or the local directory. The format of the `STATION0.HYP` file is described in this manual in the section on the HYPOCENTER algorithm (7.1). HYPO71 supports 13 test variables that influence how the program goes about locating the earthquakes. The default values for these variables were developed for the large and closely spaced networks in central California. These variables are defined at the start of the `STATION0.HYP` file by the values of TEST(01) to TEST(13). Brief definitions for each of these variables can be found below and full definitions can be found in the HYPO71 manual.

SEISAN constructs a HYPO71 format input file called `hypo71.input`, containing the station co-ordinates, thickness and velocity for each layer of the crustal model and phase arrival times, then runs the HYPO71 algorithm. The HYPO71 program generates a single output file called `hypo71.output`. SEISAN reads the information contained in this output file to create two further output files: `hypo71.out`, a Nordic format phase readings file containing the calculated location; and `hypo71.brief`, a summary file containing origin time, epicenter, depth, magnitude and station residuals.

There are a number of limitations to the current version.

- The program is designed to run from eev and can only be used for one event at a time; there is no facility for multiple event or batch location/relocation.

- HYPO71 is not included with the UPDATE command, so the database cannot be updated.

- Errors will result if the input phase readings contain arrivals from two different days, i.e. either side of midnight.

- All stations must have the same sign of latitude or longitude, so if stations extend across the Greenwich meridian and/or the equator and an offset should be added to allow for this.
Running the program

HYPO71 is run from within eev by typing hypo71, at the command line. On successful completion, the information from the hypo71.brief file is displayed on the screen. Below is an example of the screen output.

EXAMPLE RUN

# 29 13 May 2001 8:26 59 L 55.1020 -3.6388 12.3 H 2.9L 16 ?
# 29 13 May 2001 8:26 59 L 55.1020 -3.6388 12.3 H 2.9L 16 ? hypo71
HYPO71 completed successfully
Date : 13/05/01
Origin time : 8:26:59.78
Epicentre : 55- 5.45 deg N 3-37.11 deg W ( 55.0908 / -3.6185 )
Grid Ref : 296.716 East / 578.581 North
Depth : 2.13 Quality : B B*B

Statistics : 22 10 127 1 0.29 0.8 1.8
Magnitude : 2.8 ML (from 6 readings)
Magnitude : No valid coda readings

<table>
<thead>
<tr>
<th>STN</th>
<th>DIST</th>
<th>AZM</th>
<th>AIN</th>
<th>P-RES</th>
<th>P-WT</th>
<th>S-RES</th>
<th>S-WT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BWH</td>
<td>9.7</td>
<td>346</td>
<td>43</td>
<td>0.08</td>
<td>7.22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BHH</td>
<td>25.6</td>
<td>89</td>
<td>43</td>
<td>-0.03</td>
<td>1.03</td>
<td>-0.29</td>
<td>0.51</td>
</tr>
<tr>
<td>BCC</td>
<td>26.8</td>
<td>108</td>
<td>43</td>
<td>0.26</td>
<td>0.71</td>
<td>0.01</td>
<td>0.36</td>
</tr>
<tr>
<td>GCD</td>
<td>32.7</td>
<td>219</td>
<td>43</td>
<td>-0.40</td>
<td>1.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ECK</td>
<td>32.8</td>
<td>72</td>
<td>43</td>
<td>-0.28</td>
<td>1.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ESK</td>
<td>36.4</td>
<td>46</td>
<td>43</td>
<td>-0.39</td>
<td>1.00</td>
<td>-0.91</td>
<td>0.26</td>
</tr>
<tr>
<td>BBH</td>
<td>44.2</td>
<td>84</td>
<td>43</td>
<td>-0.29</td>
<td>0.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BBD</td>
<td>46.1</td>
<td>149</td>
<td>43</td>
<td>0.20</td>
<td>0.54</td>
<td>0.01</td>
<td>0.36</td>
</tr>
<tr>
<td>BDL</td>
<td>54.1</td>
<td>126</td>
<td>38</td>
<td>0.34</td>
<td>0.70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BTA</td>
<td>63.2</td>
<td>109</td>
<td>38</td>
<td>0.42</td>
<td>0.69</td>
<td>1.04</td>
<td>0.09</td>
</tr>
<tr>
<td>CKE</td>
<td>65.0</td>
<td>150</td>
<td>38</td>
<td>0.27</td>
<td>0.54</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XAL</td>
<td>93.4</td>
<td>106</td>
<td>38</td>
<td>0.16</td>
<td>0.54</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDI</td>
<td>96.6</td>
<td>16</td>
<td>38</td>
<td>0.33</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GIM</td>
<td>104.4</td>
<td>212</td>
<td>38</td>
<td>0.02</td>
<td>1.35</td>
<td>0.36</td>
<td>0.87</td>
</tr>
<tr>
<td>WCB</td>
<td>200.0</td>
<td>198</td>
<td>30</td>
<td>-0.91</td>
<td>0.45</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CWF</td>
<td>302.7</td>
<td>150</td>
<td>30</td>
<td>-0.08</td>
<td>0.24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Phase names

Only single character phase names are supported, denoted by P or S.

Weighting

Two weighting options may be used.

1. User specified weights assigned by a single integer value in the range 0 to 4 for a given phase. These will assign a weighting factor of 1, 0.75, 0.5, 0.25 or 0.0 to that phase. Also, a weighting of 9 will
assign the absolute time a weighting of 0.0 but will allow the use of relative times if a valid S-arrival is found for that station. The relative arrival time will be assigned the weight of the S-phase.

2 Distance weighting as given by the relationship \( w = (x_{far} - \Delta) / (x_{far} - x_{near}) \). By default the parameters \( x_{near} \) and \( x_{far} \) are read from the `STATION0.HYP` file. However, they can also be defined in the s-file and are used if RESET TEST(107) is set to 1. The parameters are specified in the s-file by a type-3 line, e.g.

\[ \text{XNEAR 150.0 XFAR 300.0 SDEP 7.5} \]

Using a starting location

The user can specify the use of a starting depth and epicenter by entering the character ‘S’ in columns 44 and/or 45 respectively, in the header line of the input readings file. The starting depth and epicenter are given by the values in the header line of the readings files. Otherwise, the starting epicenter is set to be the latitude and longitude of the station with the earliest P-arrival.

Fixing the location Using the character ‘F’ instead of ‘S’ in columns 44 and 45 of the header line fixes the depth and/or epicenter to the values given in the header line.

Errors

The standard error output from the HYPO71 program is contained in an additional line in the Nordic format readings output, hypo71.out, defined by the characters ‘83’ in columns 79 and 80.

The HYPO71 error line format is defined as follows:

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-14</td>
<td>A13</td>
<td>‘HYPO71 errors’</td>
</tr>
<tr>
<td>19</td>
<td>A1</td>
<td>Location quality, Q</td>
</tr>
<tr>
<td>21-23</td>
<td>A1*A1</td>
<td>QS and QD rating</td>
</tr>
<tr>
<td>25-27</td>
<td>I3</td>
<td>Number phases used</td>
</tr>
<tr>
<td>28-30</td>
<td>I3</td>
<td>Distance to closest station</td>
</tr>
<tr>
<td>32-34</td>
<td>I3</td>
<td>Azimuthal gap</td>
</tr>
<tr>
<td>36</td>
<td>A1</td>
<td>‘1’. (Always output?)</td>
</tr>
<tr>
<td>38-41</td>
<td>F4.2</td>
<td>RMS</td>
</tr>
<tr>
<td>43-46</td>
<td>F4.1</td>
<td>ERH (km)</td>
</tr>
<tr>
<td>48-51</td>
<td>F4.1</td>
<td>ERZ (km)</td>
</tr>
<tr>
<td>79-80</td>
<td>A2</td>
<td>‘83’</td>
</tr>
</tbody>
</table>

RMS is defined as \( \sqrt{\sum R_i^2 / N} \) where \( R_i \) is the time residual at the \( i^{th} \) station. ERH is the standard error in the epicenter in km given by \( \sqrt{SDX^2 + SDY^2} \), where SDX and SDY are the standard errors in latitude and longitude. ERZ is the standard error in the focal depth in km. The location quality, Q, is a measure intended to indicate the general quality of the solution and is defined by a single character.

<table>
<thead>
<tr>
<th>Q</th>
<th>Epicenter</th>
<th>Focal Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>excellent</td>
<td>good</td>
</tr>
<tr>
<td>B</td>
<td>good</td>
<td>fair</td>
</tr>
<tr>
<td>C</td>
<td>fair</td>
<td>poor</td>
</tr>
<tr>
<td>D</td>
<td>poor</td>
<td>poor</td>
</tr>
</tbody>
</table>

Q is taken as the average of QS and QD, where QS is a statistical measure of the solution and QD is rated according to the station distribution.
<table>
<thead>
<tr>
<th>QS</th>
<th>RMS (S)</th>
<th>ERH (km)</th>
<th>ERZ (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>&lt; 0.15</td>
<td>≤ 1.0</td>
<td>≤ 2.0</td>
</tr>
<tr>
<td>B</td>
<td>&lt; 0.30</td>
<td>≤ 2.5</td>
<td>≤ 5.0</td>
</tr>
<tr>
<td>C</td>
<td>&lt; 0.50</td>
<td>≤ 5.0</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Other</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>QD</th>
<th>N</th>
<th>Gap</th>
<th>DMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>≥ 6</td>
<td>≤ 90</td>
<td>≤ Depth or 5 km</td>
</tr>
<tr>
<td>B</td>
<td>≥ 6</td>
<td>≤ 135</td>
<td>≤ 2*Depth or 10 km</td>
</tr>
<tr>
<td>C</td>
<td>≥ 6</td>
<td>≤ 180</td>
<td>≤ 50 km</td>
</tr>
<tr>
<td>D</td>
<td>Other</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Magnitude

Both duration and amplitude can be used to calculate magnitudes as with HYPOCENTER (see above for details). Duration, amplitude and period for each station are used to give a magnitude value for each station. These values are averaged to give the event magnitudes.

The test variables
7.3. THE HYPOINVERSE PROGRAM, HYPINV

<table>
<thead>
<tr>
<th>Test Variable</th>
<th>Default Value</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST(01)</td>
<td>0.1 S</td>
<td>TEST(01) is the cut-off value below which Jeffreys’ weighting of residuals is not used. It should be set to a value approximately equal to the overall timing accuracy of P-arrivals in seconds.</td>
</tr>
<tr>
<td>TEST(02)</td>
<td>10 km</td>
<td>For each iteration, if the epicentral adjustment is greater than TEST(02), this step is recalculated without focal depth adjustment. TEST(02) should be set to a value approximately equal to the station spacing in km.</td>
</tr>
<tr>
<td>TEST(03)</td>
<td>2.</td>
<td>Critical F-value for the stepwise multiple regression. TEST(03) A value between 0.5 and 2 is recommended.</td>
</tr>
<tr>
<td>TEST(04)</td>
<td>0.05 km</td>
<td>If the hypocentral adjustment is less than TEST(04) then Geiger’s iteration is terminated.</td>
</tr>
<tr>
<td>TEST(05)</td>
<td>5.0 km</td>
<td>If the focal depth adjustment, DZ, is greater than TEST(05), DZ is reset to DZ/(K+1), where K= DZ/TEST(05). TEST(05) should be set to a value approximately half the range of focal depth expected.</td>
</tr>
<tr>
<td>TEST(06)</td>
<td>4.</td>
<td>If no significant variation is found in the stepwise multiple regression, the critical F-value, TEST(03) is reduced to TEST(03)/TEST(06) and the regression is repeated.</td>
</tr>
<tr>
<td>TEST(07)</td>
<td>-0.87</td>
<td>Coda magnitude constant a, where (Mc = a + blog10(T) + c\Delta + \Delta M)</td>
</tr>
<tr>
<td>TEST(08)</td>
<td>2.0</td>
<td>Coda magnitude constant b.</td>
</tr>
<tr>
<td>TEST(09)</td>
<td>0.0035</td>
<td>Coda magnitude constant c.</td>
</tr>
<tr>
<td>TEST(10)</td>
<td>100 km</td>
<td>If the latitude or longitude adjustment (DX or DY) is greater than TEST(10) then DX is reset to DX/(J+1), and DY is reset to DY/(J+1), where J=D/TEST(10), D being the larger of DX or DY.</td>
</tr>
<tr>
<td>TEST(11)</td>
<td>8.0</td>
<td>Maximum number of iterations in the hypocentral adjustment.</td>
</tr>
<tr>
<td>TEST(12)</td>
<td>0.5</td>
<td>If the focal depth adjustment (DZ) would place the hypocenter in the air, the DZ is reset to DZ= -Z * TEST(12), where Z is the focal depth.</td>
</tr>
<tr>
<td>TEST(13)</td>
<td>1.0 km</td>
<td>Parameter for auxiliary RMS values</td>
</tr>
</tbody>
</table>

7.3 The Hypoinverse program, HYPINV

The latest version of Hypoinverse-2000 program (version 1.40, 2014) has been implemented in SEISAN to be use within EEV and as standalone. The original program has not been changed except to increase number of stations (25k). The original manual is found in INF. The main program has been given the
name HYPINV (original version was HYP in conflict with HYP for HYPOCENTER) and can be run according to the original manual [Klein, 2014] and will not be described here. With a new driver program HYPINV_SEISAN, Hypoinverse can be run using Nordic files as input and output. Only the archive format for input and output is used directly with Hypoinverse.

The program does not work well at large distances ( > 1000 km) so use it only for local earthquakes.

If original data, station and control files are available, it is just typing HYPINV and the program will run according to the manual. If none of these files are available, they can be made with the conversion programs, see below.

HYPINV can be operated in 3 ways:

Alternative 1: Manually do all steps:

1: Convert a CAT file a Hypoinverse input file with the program NORHIN, e.g. norhin collect.out. The input file in Nordic format is converted to a file norhin.out in Hypoinverse format. The following limitations and additions apply:

- Only local events with at least 3 stations will be converted.
- Since HYPΟINVERSE only use first arrivals, all P and S phases will be converted to P or S, like PG to P. There is then the possibility that there will be e.g. two P’s if the original data has e.g. Pg and Pn.
- The S-P time cannot be used by HYPΟINVERSE. The weigh 9 will result in zero weight on the P but the corresponding S-time will be used so it should manually be weighed out.
- Lg and Rg phases are not used.
- Polarity and coda length are transferred.
- The hypocenter location is transferred to the norhin.out file but not the magnitudes.
- A fixed depth in input file is transferred to norhin.out. HYPΟINVERSE cannot fix depth at zero, must be at least 0.01 km. If put to 0.000, depth is not fixed. If event qualifier is P or E, the depth is fixed in output file to 0.01 km.
- In order to recover the original component names, the two letter component code is put in at available places in the archive format (columns 87 and 109). This information is written out again by Hypoinverse.
- The event id (minus year) is saved on the hypoinverse header line in the field for id. It is later used for conversion back to Nordic format.

2. Make the control files with the program MAKEHIN. This creates the instruction file hypinst, station file hypinv.sta and model file hypinv.mod. These files are standard HYPΟINVERSE files. The information is taken from the STATION0.HYP file in either the working directory or DAT. MAKEHIN cannot work with an alternative STATIONx.HYP file. If the hypinv.sta or hypinst files are present, they will not be generated. This allows for changes in the Hypocenter parameters to be used in subsequent runs. On the other hand, if the STATION0.HYP file is changed, the hypinst and hypinv.sta files must be deleted before the changes in STATION0.HYP will be used. The following parameters are partly generated from STATION0.HYP:

- Vp/Vs
7.3. THE HYPOINVERSE PROGRAM, HYPINV

- Start depth
- Minimum number of stations, however less than 3 is not allowed.
- Distance weighting starts at iteration 4, starts at HYPOCENTER parameter xmin and is zero at approximately xfar. However that assumes that xmin is further away than the second closest station. Hypoinverse parameter DISCUT is set to xmin, DISW1 is set to 1, and DISW2 = xfar/xmin. See Hypoinverse manual for details.

3. Type HYPINV and the program runs. There is a one-line output per event on the screen and the full output is in a file called print.out.

The output file from HYPINV is now hypinv.out and it can be converted back to Nordic format with program HINNOR, however, some of the original information is lost (like e.g. amplitudes). To avoid this, see alternative 2. HINNOR transfers the following information:

- Hypocenter with rms.
- Error line with gap and erz (vertical error) and erh (horizontal error). The Nordic values erlat and erlon are both replaced by erh.
- Magnitudes L and C in Hypoinverse file. If program HYPIN_SEISAN is used, the magnitudes are overwritten with the original values.
- Residuals.
- Distances, Hypinvers will give 999 if more than 1000 km.
- Angle of incidence.
- Azimuth.
- Weights used.
- Polarity.
- Coda length.

Alternative 2: Run the whole process using the driver program HYPINV_SEISAN. The program does the following:

- Run NORHIN
- Run MAKEHIN
- Run HINNOR
- Generate an output file in Nordic format hypinv_seisan.out that contains the new HYPINV location with corresponding residuals etc as well as all the original information in the Nordic input file like amplitudes, spectral information etc. The only thing lost is the original P and S phase names. All the R and D events skipped by NORHIN are put back in. To keep track of the events, the ID is used. The ID line will get the action flag HIN. The original magnitudes, agencies and number of stations are kept.
Using alternative 2 makes it very easy since Nordic file is both input and output.

Alternative 3: Running HYPINV from EEV using the command H. The steps are:

- EEV runs HYPINV_SEISAN using current s-file.
- The print.out file is printed on the screen.
- The user is asked if the data base should be updated with the combined file hypinv_seisan.out.

It is thus possible to use HYPINV to update the data base of local events with the limitations mentioned under alternative 1.

Note: Hypoinverse use residual weighing by default so rms and errors might be smaller than locating with Hypocenter.

Magnitudes

No magnitudes are transferred from Nordic files to Hypoinverse files. Coda magnitude is calculated with Hypoinverse using the Hypoinverse default relations. If the magnitudes have to be updated due to significantly changed location, it must be done with SEISAN EEV command 'u' and with the hypocenter fixed so the original HYPINV location is preserved.

How to run Hypoinverse with other parameters

Hypoinverse has many options of which the multiple models might be the most interesting. Generate the standard input files as in alternative 1 and then modify them according to the Hypoinverse manual. Then run HYPINV directly or, as the hypinst file is not changed if it exists, using alternative 2 or 3.

Program written by Richard Luckett

ISC has for many years used a standard procedure to locate earthquakes and the ISC locations have often been used as a reference. The earth model used is the Bullen tables. ISC has recently rewritten the old location program and it was therefore possible to also port it to SEISAN. The purpose is that it should be possible to compare standard ISC locations with location using other programs and models. The implementation in SEISAN was done using the standard hyp program where only the location routines have been changed. The program then behaves almost identical to HYP and uses the same format input and output files.

Parameter files: STATION0.HYP is used for station coordinates, magnitude scales and agency code. The crustal model information is not used and only the RESET TEST parameters related to magnitude are used. In addition, there is a new parameter file (in DAT) iscloc.def with parameters specific for the ISC location routines, see file for explanation of parameters.

Input data files: Just like for HYP

Output files: Hyp.out is like before, print.out is different.

Not all crustal phases used with HYP may be available. The weights used in SEISAN do not apply since the program uses residual weighing only, see parameter file.

Magnitudes are calculated exactly like in SEISAN.

In eev, the command to locate with HYP.ISC is 'il'.

For more information about the ISC location program, see http://www.isc.ac.uk/Documents/Location/
7.4 Plugin for the iLoc location algorithm, ILOC_SEISAN (Linux only)

ONLY IN ALPHA VERSION AND ONLY TESTED ON LINUX

A plugin for version 2.04 of the iLoc program, written by István Bondár [Bondár and Storchak, 2011], has been implemented in SEISAN to be used within EEV or as stand-alone. The purpose of the plugin is to have the possibility of event location with a regional 3-dimensional velocity model. Since iLoc use the Regional Seismic Travel Time (RSTT) [Myers et al., 2010] velocity model as part of its event location it serves this purpose.

iLoc must be installed on the system used and the program iloc must be in the path. One only need to compile iLoc with the ISF I/O option to be used with SEISAN. RSTT is part of iLoc and is required. See the iLoc installation instructions: [http://www.seismology.hu/index.php/en/home/iloc](http://www.seismology.hu/index.php/en/home/iloc)

Note, only stations registrated to the ISC are included in iLoc. Otherwise one must add used stations in the file isc_stalist in the iLoc aux data folder. If your stations have been registrated at the ISC recently, download a new station file at: [ftp://ftp.isc.ac.uk/pub/iscloc/isc_stalist](ftp://ftp.isc.ac.uk/pub/iscloc/isc_stalist)

If iLoc does not have the station information it will type: textttReadISF: missing station STATIONCODE

There are three ways of using iLoc in SEISAN:

1 Stand-alone, type:
   iloc_seisan <sfilename> <operator_id>
   The program ILOC_SEISAN will read the sfile, convert the sfile to isf format, using the ISFNOR program (output filename: norisf.out). Next the iLoc program is executed, it reads the isf file. If data are not ok for hypocenter inversion iLoc will end. If data are ok for hypocenter inversion result is printed on the terminal and written as isf format in the file iloc-out.isf.

2 From EEV; in EEV type:
   iloc
   The process here is similar to the the stand-alone, but the user will have the option to save the output in the sfile.
   Below is the first five header lines of an sfile that includes the iLoc hypocenter solution. The two last type 3 lines are from iLoc:

   2016 0229 2320 01.8 LE 67.177 20.623 0.1F UPP 9 0.2 1.5LUPP 1
   2016 0229 2320 02.6 DE 67.168 20.600 0.0F HEL 17 0.3 1.2LHEL 1
   LOCALITY: 612636992 Sweden 3
   2016 0229 2320 02.67 67.1608 20.6182 0.0 f 0.45 INTAR ILOC 3
   GAP 79 E 0.98 j 4.2 i 3.9 A 82 s 16 d 26m 0.04M 3.87 INTAR ILOC 3

3 In a loop over many events:
   Make a index.out file with SELECT program, run the program ILOC_LOOP. ILOC_LOOP will read all sfiles in index.out and add the iLoc solution to the sfiles if a solution was found.

The format of the first type 3 line is the same as the type H line described in Appendix X. The format of the following comment line is:

1:4 ' GAP'
5:7 Gap in azimuthal coverage (degrees), i3
8:9 'E'
10:14 Origin time error (seconds) f5.2
15:16 'j'
17:21 Semi-major axis of 90% ellipse or its estimate (km), f5.1
22:23 'i'
24:28 Semi-minor axis of 90% ellipse or its estimate (km), f5.1
29:30 'A'
31:33 Strike of error ellipse clock-wise form North (degrees), i3
34:35 's'
36:39 Number of defining stations, i4
40:41 'd'
42:45 Number of defining phases, i4
46 m
47:52 Distance to closest station (degrees), f6.2
53 M
54:59 Distance to furthest station (degrees), f6.2
60 Free
61:65 Agency, a5
66 Free
67:73 'ILOC Method used, a7
74:79 Free
80 '3'
Chapter 8

Trace plotting, phase picking and spectral analysis, MULPLT

This program is the general plotting and signal analysis program. The program is capable of doing general phase picking, correct for instrument response, and produce Wood-Anderson seismograms for determining MI, synthetic traces for Mb and Ms, determine azimuth of arrival for 3 component stations, do spectral analysis and particle motion. The program can also read in theoretical arrival times for global phases for help in identifying phases. If a quick location is needed based on a waveform file only, mulfplt can both pick the phases and locate the event. MULPLT operates either as a database independent program (started with command MULPLT) or in connection with the database (started from EEV with command P or PO). If the program works independently of EEV, it will create an output file mulfplt.out in Nordic format with the readings and results of spectral analysis. This file can directly be used with e.g. HYP. MULPLT reads and plots one channel at a time. This can be very time consuming when reploting traces and from SEISAN version 8.2, a certain number traces are kept in a memory buffer in order to speed up reploting and reprocessing. The data is stored in one large array the size of which is determined at the time of compilation so for systems with many or long traces, it might have to be larger and for systems with little memory it might have to be smaller. The dimension is set in file seidim.inc in directory ./INC using variable max_mem_sample. A typical value is 30 000 000.

Note on Seed/Miniseed reading

The routines will put in zeros for gaps
If an overlap occurs, the block with the overlap will be discarded and zeros inserted.
If a block is out of time sequence, the block will be discarded, this is like a large negative overlap.
The channel start time will be considered the time of the first block found for that channel. If a block then occurs before the start time the block will be discarded.

Starting MULPLT from prompt line

Giving the command mulfplt -h, will show the basic options:

The MULPLT program - for plotting earthquake data

Usage: mulfplt [options]
144 CHAPTER 8. TRACE PLOTTING, PHASE PICKING AND SPECTRAL ANALYSIS, MULPLT

enter input

## Options ##
- help          Print this list
- h             Same as -help
- plotdefault   Will plot data directly with default values
- qdp           Same as -plotdefault
- po            Same as -plotdefault
- version       SEISAN version

Examples:
  1) mulplt
     enter values
  2) mulplt -plotdefault -sfile /home/seismo/REA/TEST_/1996/06/03-1955-40D.S199606

Giving command mulplt, the question is

Filename, number, filenr.lis (all)
Continuous SEISAN data base: cont
Large SEED volume: conts
Archive: arc
Make a choice

Filename, number, filenr.lis(all): The program asks for a file name or file number of a waveform file. To use the number, it is assumed that a list of files has first been created and numbered in a file filenr.lis using command DIRF, see section 15. By giving the number, the file corresponding to the selected number is used. By giving a ?, the list with numbers is displayed and a new number can be given. If many files are to be plotted with one command (hard copy only), give filenr.lis for file name and all events in FILENR.LIS will be plotted. There will only be one question about filter and then all events are plotted with all channels and the chosen filter.

Cont: Plot from a continuous data base. The program will use all data bases defined in SEISAN.DEF. A question will be given for absolute start time and window length. See also 8.5. Conts: Plot from a large SEED volume. A SEED file too large to be read in can be plotted in parts. A question will be given for file name or number. The file is then read and available data is displayed. Start time and window length is then entered.

arc: Plot from a BUD or SeisComp archive
Plot from a continuous archive. The program will use all channels defined in SEISAN.DEF. A question will be given for absolute start time and window length. See also 8.5. Then follows a display showing all stations and virtual networks that are defined in SEISAN.DEF before a channel selection is made. It is possible to combine selection of stations with one or more virtual networks. MULPLT removes duplicates in case a station should be selected more than once. See also section ‘Working with many channels in MULPLT’ a bit later.

8.1 MULPLT main functions

The program has 7 main functions irrespective of type of input as illustrated below with the questions given by the program:
8.1. MULPLT MAIN FUNCTIONS

<table>
<thead>
<tr>
<th>Plot options:</th>
<th>Interactive picking</th>
<th>Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi trace plot on screen, def (0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi trace plot on screen (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi trace plot on screen+laser(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi trace plot on laser (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous on screen (4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous on screen + laser (5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous on laser (6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stop (9)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Return: Picking phases, spectral analysis and 3 component analysis

0: Initial plotting of new waveform data and registration in database using predefined defaults. Phase picking.

1: Initial plotting of new waveform data and registration in database, or general plotting of multi-trace data. Phase picking. In this mode, the user is asked to select which channels to work with and a graphical window will be shown for selection. If more than 250 channels are available, selection will have to be made on several screens. A maximum of 1000 channels can be used. Optionally, channels can be shown in alphabetical order (see MULPLT.DEF).

2: Same as 1, only hardcopies may be made at the same time.

3: Making hardcopies of many waveform files with one command. No screen output.

4,5,6: Plotting one channel continuously like on a seismogram with several traces from left to right on top of each other. One channel can be selected. Time windows can be selected for event extraction.

For continuous data, see 2.2.2 and 8.5.

Commands f and q: With a plot on the screen, q will always quit mulplt. F will, in single trace mode, bring up the next channel, in multi trace mode bring up the next event whether input is from a filenr.lis or EEV.

Option 0 is particular useful for checking many new events, since the program does not ask question about station choice (uses definition in MULPLT.DEF file in DAT, or working directory) and typing f (when the plot is on the screen) automatically goes to the next file in filenr.lis.

If option 1,2 or 3 is used, a display will show the available channels and the user can click to select. If MULPLT is operated from EEV, a * in a channel box will indicate that readings are available. NOTE: The channel will be marked if only the orientation code is matching. When the channel selection is shown, it is possible to quit the program with q or continue with f.

If option 4,5 or 6 is selected, continuous data is plotted, see below.

Running MULPLT using command MULPLT, the program asks for a file name or file number of a waveform file. To use the number, it is assumed that a list of interesting files has first been created and numbered in a file filenr.lis using command DIRF, see section 15. By giving the number, the file
corresponding to the number is used. By giving a ?, the list with numbers is displayed and a new number can be given. If many files are to be plotted with one command (hard copy only), give filenr.lis for file name and all events in FILENR.LIS will be plotted. There will only be one question about filter and then all events are plotted with all channels and the chosen filter.

**Hardcopies assume a PostScript printer.** For each event plotted, a plot-file called mulplt.eps is generated. The plot files are sent directly to the printer from within the program with the seisan print command as soon as the plot is finished for one event but before the program is finished. In Unix, this is lpr or lp while on PC, the command is given in a .bat file in COM (see installation section). This means that the same plot-file is overwritten for each event plot. For setting up the printer, see installation, chapter 3.

In multitrace mode, many traces (number limited by the SEISAN system definitions, see chapter 3) can be plotted. If the plot is made via EEV, all picks are also displayed.

**Location of waveform file:** Mulplt will search in current directory first. If not found there, the WAV directory is searched. If the SEISAN.DEF file has been set up, MULPLT will thereafter try to locate the waveform file in one of the databases or specific directory given in the SEISAN.DEF file (located in DAT or working directory).

Format of waveform files: MULPLT can, on all platforms, use SEISAN, GSE, SAC ASCII and binary, GURALP (one channel files), Helmberger Format and SEED. If started from EEV, files in different formats can be used at the same time.

**Time gaps in waveform files:**
Only SEED format has running time headers and therefore the only format where a possible time gap, in one file, can exist. SEISAN will replace missing data intervals with zeros. For a SEISAN continuous data base, it is assumed that a gap of less than 2 s between files is not a gap. Larger gaps will be replaced by zeros.

### 8.2 Use of MULPLT from EEV and SE

In order to process events more easily, EEV and SE and MULPLT have been tightly integrated. When MULPLT is called from EEVi or SE, command f will plot the next event in the database, to go back to the same event in EEV, use quit. The event can be plotted with default parameters from EEV using PO. If PO has been selected, the f command in MULPLT multi trace mode will show the plot of the next event with default options. This is a fast way of plotting waveform files going through S-files in a database. The command poo does the same as po except that you start in single trace mode. In SE, command P corresponds to PO in EEV.

If several waveform files are available, the user will graphically be shown the files and can select one or several Files can optionally be displayed in alphabetical order (see MULPLT.DEF). If more than 75 files, several selection screens will be shown. A maximum of 1000 files can be used if the PO option has been used, all default channels will be used.

Plotting from EEV or SE, a file with a list of waveform files can also be used as input. The filename must end with _LIST. The filename could be e.g. 2002-02-02-1310-20S.BER_LIST. The format is the filenr.lis format so the file can be created with DIRF. The _LIST file can be in local directory, WAV or any other directory as specified for normal waveform files. The intention with the list option is to be able group many waveform files in one group without the need to merge the files or list them all in the S-file. The limit of number of files in the _LIST files is 1000 and up to 10 _LIST files can be used for the same event.
8.3. WORKING WITH MANY CHANNELS IN MULPLT

There is currently no automated facility to create and include .LIST files in the S-file.

Components: The picks in the S-file are plotted on respective traces if both the station name and component matches. The waveform file has in general a 3 letter component code while the S-file only has 2 so only the first and last letters are checked for matching. Similarly, when writing a pick to the S-file, only first and 3. letters are used. In case there are no component code in S-file, the readings in the S-file are shown on all components. If any subsequent reads are done, the component code of the first channel used will be assigned to all readings for that station.

8.3 Working with many channels in MULPLT

Many networks have several hundreds of channels which quickly becomes difficult to work with. MULPLT has several options to facilitate working with many channels, whether working with an archive, SEISAN continuous data base (cont base), individual files or a combination. Note that MULPLT is setup to handle up to 1000 channels. Some of the options are:

- **Limit the number of channels on one screen (all input):** The number of channels per screen is parameter NCHAN PER SCREEN in MULPLT.DEF. From each screen, channels can be selected and the user can go forward to the next screen with TAB. It is not possible to go backwards. If there is a need to temporarily see all channels on one screen, command N in multitrace mode will do that. This can be useful if the Out function (writing out a file of what is seen on the screen) is used. Alternatively, when registering an event from an archive or a cont base, the extracted file can have all channels irrespective of what is seen on the screen.

- **Set default stations to be plotted in default mode (all input):** The stations are defined in MULPLT.DEF and only those stations will be plotted in default mode.

- **Plotting only stations with readings or only Z-channels (all input):** When the channel selection screen comes up, there is an option to select only stations with readings (Picked or p). Since then channel selection screen only has 250 channels, there might be up to 4 channel selection screens and the next screen is chosen with f. If however F is pressed, only channels with readings are also selected for the remaining channels without the following screens being shown. Similarly the command for only Z-channels are z and F, respectively, to get all Z-channels for the whole data set.

- **Plotting only stations within a given distance, radius, from a user selectable point, the midpoint (all input):** Since most events with large networks are only recorded with the nearest stations, this option is very effective in limiting the chosen data to the most important stations. There are several options for selecting midpoint and radius using parameter MULPLT AREA in MULPLT.DEF:

1: Midpoint from epicenter in S-file, radius from MULPLT.DEF
2: Midpoint from epicenter in S-file, radius asked at start of MULPLT
3: Midpoint and radius from MULPLT.DEF
4: Midpoint from MULPLT.DEF, radius asked at start of MULPLT
5: Midpoint and radius asked at start of MULPLT
6: Midpoint from a station in MULPLT.DEF, radius from MULPLT.DEF
7: Midpoint station asked at start of MULPLT, radius from MULPLT.DEF
8: Both midpoint station and radius asked at start of MULPLT

In addition, in multitrace mode, the radius can be changed (command R) and a center station can be entered (command S). If command S is given and there is no radius defined, the user will also be asked for radius.

**Plotting stations from an archive:** Very many channels can be specified in the archive (currently max 3000). All channels will initially be available to the user and the start times etc read in. This can take some time, so in order to avoid first reading the channels and then selecting channels present with data, it is possible to preselect which channels should be checked (first station selection screen when using arc). A further reduction in the number of channels to check is to use the option above for distance to a station or a point. This distance check is done before the channels are checked.

Archive referencing in S-files is described in section 2.2.3. An example of an archive reference line is:

```
ARC STAT COM NT LO YYYY MMDD HMMM SS DUR
ARC BORG LHZ II 10 2011 0129 0650 00 3600 6
```

which is referring to one channel (station STAT, component COM, network NT, date-time and duration DUR (s)). With many channels, this referencing becomes a bit cumbersome and some wild cards can then be used:

Station is blank or *, component, network and location are blank: All channels defined in SEISAN.DEF will be plotted with start time and duration given in ARC line (if not blank, see next option). If a component is given, only that component will be selected. If a network is given, only that network will be chosen. If a location code is given, only channels with that location code is selected.

Start time and or durations blank: Start time will be origin time - a time given in SEISAN.DEF (ARC START), duration will be a time given in SEISAN.DEF (ARC DURATION).

Station is P: All channels for all stations listed in the S-file found in the archive will be plotted. There is no requirement for the station to have any other information than the station name, component code is not used. So a new station can easily be added.

Plotting all stations without an archive reference line: If parameter ARC BY DEFAULT in SEISAN.DEF is set to 1, all channels in the archive will be selected. Setting it to 2, only stations with readings are plotted.

**NOTE:** An ARC line can be inserted/edited in the S-file from EEV by command arc. If there is ARC line(s) in the S-file, these will determine what is plotted and not the parameters in SEISAN.DEF. If e.g. the station name in ARC line is X, no channels will be plotted (assuming there is no station with name X) and only other possible waveform file names will be used.

Optionally, the archive channel specification lines can have a time interval of validity. This is to avoid unnecessary checking of archive files if the archive has channels from different time periods. The format is: yyyymmdhhh for both start and end time, only as much info as needed has to be specified. If start time is blank, start time is very early, if end time is blank, no end time. The start end end times are written in columns 61-70 and 71-80 respectively.

**How to specify many channels in a separate file, the _LIST file:** In some formats, a waveform file can only have one channel (e.g. SAC). Referencing hundreds of waveform files (or ARC references) in an S-file is not practical. These file names can be collected in a separate file in filenr.lis format. The file name must end with _LIST like e.g. 2012-12-01-1044-22.BERGEN_LIST and the filename is referenced as a waveform file in the S-file and also stored in the usual place of the waveform files. Up to 10 _LIST
files can be referenced in the S-file so a large network can be broken up in regions, each with a _LIST file, which then can be selected separately in the file selection screen. There is no specific software, except DIRF, to create _LIST files.

8.4 Continuous plotting of one channel

This option is used to plot one channel in a multi line display and can therefore simulate a helicorder plot. This is a very different option from the plotting from a continuous data base (see 8.5). Interactive processing is not possible in this mode except for selecting time windows for event extraction, see below.

Using this option the program asks for the following input:

Low and high cut for filter: Give values or return for no filter. A band pass cannot always be used with a low frequency low cut(filter unstable) so if e.g. a LP record is to be simulated, use filter limit 0 to 0.1 Hz. The zero means it is a low pass filter, not bandpass. A filter 10 to 0 would mean a high pass filter.

Seconds pr line: Number of seconds on each line.

End time: This question only appear if plotting from EEV. The list of files is then the list of events belonging to the data base used. Give end time as e.g. 2000050203

Max count: The absolute maximum count to be used for full scale. Since many lines and possibly many pagers are plotted, it is not possible to use autoscaling, and like on a seismogram, a fixed value must be set.

Lines pr page: Number of lines per page.

Station code: Station code (max 5 characters)

Component code: The component code, max 4 characters. NOTE, since component codes are 3 characters, the first 2 and the 3. must be used. Like BH Z for BHZ. Some very old SEISAN waveform format files might have 4 characters.

MULPLT will plot from the first file given from the filenr.lis file and then continue to plot as long as more file names are given in filenr.lis. Alternatively if plotting from EEV, it will start with the current event and continue until the end time. So if a month of data files are given, a month of seismograms will be displayed. There is no requirement that the input files follow each other in time (no time gaps) since each file is plotted on the page where it belongs in time. However, the files must be time ordered. The continuous option can therefore be used to check availability and timing of continuous data. Discrete events can also be plotted in this mode if one want to get a display of when the events occurred. However, if filtering, it is assumed that the files follow each other in time since a few points are carried over from one file to the next to make the filtering continuous. Figure 8.5 shows an example.

Selecting time windows for event extraction in continuous one channel mode:

To extract events shown in the continuous plot one can mark a given time window by typing ‘s’ and ‘e’ at the start and end of the required time window (see Figure 8.4). Selected time windows are, when plotting is finished, written to the file mulplt.ext and to the screen as wavetool command lines. By executing mulplt.ext (type sh ./mulplt.ext or source mulplt.exe in UNIX, on PC the name will be mulplt.ext.bat so just typing the name mulplt.ext will start the extract process) the time windows given in mulplt.ext, are extracted from the continuous database defined in SEISAN.DEF. The data files are extracted in the local directory. These files can now be registrated in a REA database with autoreg. One cannot delete a “Start” or “End” time mark. So, if another time window is required pick the new window and delete the line in mulplt.ext with the old time window before data is extracted.
The time marks is written to the file mulplt.ext for data to be extracted and processed. Type “s” and “e” to add time marks.

If the SEISAN_LOGGING parameter is set to 1 in SEISAN.DEF the analysis done with this option, will be saved in a log file in REA/BASE/_LOG/YEAR/MM. BASE_ is the default database, YEAR and MM is the year and month the analysis was done. The log files are named mulplt.cont.data-time.log, where date-time is the time of the analysis. Logging is on by default.

### 8.5 Commands in MULPLT, overview

When the trace(s) are on the screen and the cursor is displayed, then several options are available. Most options can be displayed by pressing the MENU button in the upper right hand corner. Pressing MENU again removes the option boxes. Commands can be given by either pressing a letter or clicking on a box in the menu (Figure 8.6). By pressing ? or clicking on the Help button, the following help menu will be displayed:
8.5.  COMMANDS IN MULPLT, OVERVIEW

Help on MULPLT

MULPLT has 3 modes:

  multi trace mode: Displaying many traces, some picking options
  single trace mode: One trace, all picking options
  all channel mode: All channels for selected stations

Most commands are given by pressing one key, however several commands can also be given by clicking in the appropriate menu box on top of the screen. If the full menu is not displayed, select the menu on top right part of the screen. The following commands are available, first given by the single letter command and if also available as a mouse click, the letters in the menu box is shown. COMMANDS ARE CASE SENSITIVE.

Channel selection: In multitrace mode, one or several channels can be selected by clicking on the station code. Several channels can be selected by first clicking on first channel with left mouse button and then clicking on last chanel with right mouse button. If only one channel is selected, and toggll (t) to single trace mode is done, a new toggll in single trace mode will display all previous channels again. For the channels selected in multi channel mode, it is possible to change to single station 3-component mode with y and back again with y. If all channels from all stations shown in multi trace mode are desired, press u, back again with u or y.

From single channel mode it is also possible to go to all channel mode for that station with y and back again with y.

Two channels can be plotted on same trace with different colors. If the channel below is to be plotted on top the one above, the channel below is selected by putting cursor on channel and pressing & and then r for repolot.

Zooming:  Select a window with the mouse and a zoomed window will appear below in single mode and replace the plot in multi mode. In Single mode, it is also possible repolot the zoomed window in place of the original by placing the cursor above the trace when selecting the zoom window. This makes it possible to zoom in the zoomed window. In order to go back to the original window in multi mode, do an opposite zoom, meaning picking the last point first.

a:  Read amplitude. Position cursor at the top of a wave and press a. Position cursor at the bottom of the wave and press a. Amplitude(0-p) and period are now stored. These values will be stored with the amplitude phase selected. Amplitude and period are displayed.

A:  Automatic amplitude reading.
b: Filter 5-10 hz, see below

B: Back Go back one trace in single trace mode
   From eev, multi: Go back one event
   Continuous data base: Go back one window

c: Read coda.

C: Read end of coda automatically

d: Delete phase. Position cursor near phase and press d.

d: Del W: Delete waveform file(s), cursor outside area of trace plot, file(s)
   must be in working directory, else no files displayed. If done
   from EEV, only file names in S-file are deleted.

D: Del S: Delete S-file if operated from EEV, multi mode

e: Phase E

E: Call external program (defined in MULPLT.DEF) to compute spectrogram.
   Only in multi trace mode

f: Next: Single: Go to next channel
   Multi, One event: Go to next event
   Multi, continuous data base: Go to next window

F: FK: FK analysis of array data

g: Ground: Make a ground motion seismogram(s).

h: Azim: Make 3 component analysis (single mode ONLY) to
   determine azimuth of arrival. Select a window around
   the P-phase on the Z-component. Azimuth and apparent
   velocity will enter the S-file with the next phase
   reading.

i: Phase I

I: Iasp Calculate IASPEI synthetic arrival times, which are then
   displayed, multitrace only. For teleseismic events
   the IASP91 tables are used, while for local events
   the model defined in STATION0.HYP is used.
   If I is pressed again the synthetic arrivaless are removed
   and the iasp.out file is deleted.

j: mb: Generate a synthetic SP seismogram for reading amplitudes
   for determining mb.
J: mB: Generate a synthetic BB velocity seismogram for reading amplitudes for determining mB.

k: Ms: Generate a synthetic LP seismogram for reading amplitudes for determining Ms.

K: MS: Generate a synthetic BB velocity seismogram for reading amplitudes for determining MS.

l: Locat: Locate event, only multi trace mode

m: Filter 15-24 hz, see below

M: Merge: Merge waveform file, mulplt called from EEV, only if files are in working directory.

n: Filter 10-15 hz, see below.

N: Toggle multiple windows to show all traces in one screen or one of the multiple windows.

o: Oth C: Select other channels

O: Out: Makes an output waveform file of current data on screen. Only multi mode. Response info not saved. Standard name with YYYY_MM... is used, name is written in text window.

p: Regis: PUT (Register) event in database, you will be prompted for event type and waveform file will be copied to WAV.

P: PartM Particle motion plot, requires the three components in multi trace plot

q: Quit: Quit program

r: Plot: Replot same event, useful when screen is cluttered up with e.g. many picks. Also used when a replot is wanted with new parameters e.g. filter.

R: Change radius from epicenter or other selected point from which to select stations.

s: Spec Make a spectrum, single mode ONLY. Press s, select window with cursor, answer questions and the spectrum will appear below with noise spectrum.
s: Make a spectrogram of all traces, multi trace mode only.
S: Same as s without the noise spectrum, single trace.
S: Select station for area selection, only multi trace mode.

T: Toggle: Toggle between multi and single mode

T: OutW: Output file multplt.wav of what is on screen

u: Same as 'y' with the difference that all components for all stations are shown.

U: Rotate: Rotate components. Backazimuth is calculated from location in header or as a second option, taken from observed + azimuth residual for respective station. R is plotted instead of N and T instead of E. Combining 3 component option (h) with u, the user is asked for the backazimuth angle.

v: Filter 1-5 Hz, see below.

y: AllC: Toggle to and from all channel mode. All channels are only for one station, for more stations use 'u'.

Y: Pick a theoretical phase if displayed. Place cursor where the phase might be. Press y and the program will select nearest synthetic phase.

w: WA: Remove system response and display synthetic Wood-Anderson ground motion in nanometers (nm) on next plot (using R or zoom).

W: Oth W: Select other waveform file, same event

z: Filter 0.001 to 0.1, see below
x: Filter 0.1 to 1.0 Hz, see below
Z: <W> Increase window length in plotting from a continuous data base
X: >W< Decrease window length when plotting from a continuous data base

>: Print: Will make a hardcopy of all channels of current event with the last selected filter, only in multitrace mode.
<: Same as D

*: Scale: Fixed scaling of trace amplitudes.

Up and down arrows: Increase and decrease amplitude

Horizontal arrows: Scroll
_: Dist: Select plotting channels in distance order.

: Resp: Plot response file, single trace mode only.

TAB: NextW: Next window if multiple windows

Filter options: The fixed filters (4/8 pole Butterworth) are placed on keys z,x,v,b,n,m with the following frequencies:

z: 01-.1: 0.01 - 0.1 Hz Pressing key once gives a 4 pole filter one way
x:.1-1.: 0.10 - 1.0 Hz Pressing the key twice, and the filter also
v: 1 - 5: 1.0 - 5.0 Hz go the other way and it is now an 8 pole filter.
; 2 - 4: 2.0 - 4.0 Hz
b: 5-10: 5.0 - 10.0 Hz From menu, only 4 pole filters are used.
n: 10-15: 10.0 - 15.0 Hz
m: 15-23: 15.0 - 23.0 Hz

.: Filt: Variable filter, question of filter limits is given in text window.

,: FixF Fix filter. If pressed after selecting as filter, the filter remains fixed until pressing ',', again.

Variable filter also with number of poles

& Overlay, see in beginning of help section.

Phase picking: This is possible in both modes. In Single mode, the phases defined are shown on top with some of the options, while they are not shown in Multi mode, but have the same definitions.

Combining options: Note that you can select several options together.

E.g. V and S will first filter the signal and then make the spectrum.

Saving observations: When you go to the next trace or another event (F), the readings are saved in the S-file. They are also saved when you pick the next trace in Multi mode.

This is also true for spectral parameters and azimuth determination.

jh 3-2-2020

Filters in MULPLT

All filters in MULPLT are Butterworth filters in time domain. When a filter is selected, using a hotkey or from the menu, the filter is only run one way, forward in time, and the number of poles is then 4. This will make a small phase delay where the first onset might appear a bit later, so if possible, read on unfiltered traces. If a 2 way filter is desired, press the filter key twice and the filter will also run backwards in time and the filter will be similar to an 8 poles filter. This gives theoretically a zero phase shift filter, however in practice, some of the onset energy is seen well before the first arrival, so it seems to distort the arrival times much more than using the 4 pole filter. When the program asks for a non
fixed filter like when using the "Filt" command, the filter is always 4 poles by default. However, it is now also possible to interactively select number of poles and number of passes (1: forwards, 2: both ways) using the ' command. Press ' and the user is asked for filter frequencies, number of poles (less than 10, but more then 4 and the filter might become unstable for high sample rates) and number of passes. In addition LP (low pass), HP (high pass) and BR (band reject) filters can be used. E.g for a 5-10 Hz filter some of the choices are:

5 10 4 pole band pass, command .
0 10 4 pole low pass, command .
10 0 4 pole high pass, command .
-5 10 4 pole band reject, command .
5 10 2 2 pole band pass, command '

WHEN PLOTTING, THE FILTER LIMITS, NUMBER OF POLES AND NUMBER OF PASSES IS WRITTEN ON THE SCREEN.

For band pass filters, the number of poles for both frequencies is the same. When doing spectral analysis or response removal and specifying a filter before, the filtering is done in time domain and the filter has the number of poles specified by the user, default 4. NOTE: When reading polarities, DO NOT USE FILTER, if possible.

Filtering and instrument correction: Since filtering is done in time domain, there is an added stability filter in frequency domain to avoid low frequency blow up. This filter is a 4 pole HP filter at 1/5 the filter low frequency corner.

Filter limitations: For frequencies below 0.5 Hz, only 4 pole BP and BR filters can be used. If the user try to select another number of poles, the number of poles is set to 4.

Filtered output: Extracting data with WAVETOOL, option 'Out'. It is only possible to use 4 pole BP filters, forward in time, using any other filter and the data is not filtered. Using option OutW any filter can be used but then only Ascii Helmberger format is possible. WAVETOOL can then convert to any other desired format.

Prior to version 9.1 MULPLT used a 4 pole Butterworth filter in time domain and an 8 pole Butterworth filter in frequency domain. The filters in frequency domain were use in connection with instrument response correction and spectral analysis. It has turned out that the frequency domain filters distorted the signal in some cases, particularly for narrow band and low frequencies. Therefore, frequency domain filters are no longer generally used. The change in filter setup, might change MI magnitudes by 0.05 to 0.1 depending on which filter (if any) was used.

Displaying epicentral or hypocentral distance

There is an option in MULPLT.DEF to plot epicentral or hypocentral distance on multi trace plots. The distance is shown just after the DC and is given in km. The option is set with parameter PLOT DISTANCE. The default is not to plot distance.

Displaying uncertain time

In each trace header in the SEISAN waveform file, there is a flag to indicate if the time might be uncertain (see Appendix B). If that flag has been set, the message ‘UNCERTAIN TIME’ will be displayed on top of the trace. Currently this flag is only put into the waveform files if the data comes from a SEISLOG system that has detected a timing error or if the data is converted from SEED/MiniSEED data. Similarly plotting SEED/MiniSeed data, uncertain time will be displayed if that flag is set in any block in the time window read in for a particular trace.
Figure 8.2: An example of using MULPLT in multitrace plot mode. Notice that start and stop times are different for different channels. The horizontal line at the start of the plot is the DC level. The small number above each trace to the right is the max absolute count with the DC-level subtracted and the small number to the left above the trace is the DC level. If plotting from EEV, the phase picks available are shown.
Figure 8.3: Examples of MULPLT with theoretical arrival times of some global phases. Short period seismograms are shown. The theoretical phases are marked with onset $y$ below the trace and the read phases are marked normally above the trace.
Figure 8.4: Example of MULPLT with theoretical arrival times showing global phases on a long period seismogram. The filter used from 0.01 to 0.1 Hz. Without filtering, almost nothing would have been seen on this broadband station.
Figure 8.5: MULPLT in continuous mode.
The plot shows 6 hours of long period data. The scale is 3000 counts between the traces and the filter used is from 0.01 to 0.1 Hz. The trace start time in hours and minutes is given on top of each trace. On the header line, P1 means the first page and DC is the DC level subtracted. Note that the numbers on the time scale at the bottom only are valid for the first trace unless all traces are 60 sec or 60 min long.
8.5. COMMANDS IN MULPLT, OVERVIEW

Below is some more detailed description of some of the options. The one letter command is given with the menu command in parenthesis:

To apply filters, first make a selection of options (filter, window, channel selection) and then execute by pressing R(Plot) (or selecting a zoom window). Figure 8.2 shows an example.

Single trace mode:
In this mode, one trace is initially displayed on top of the screen, see example on Figure 8.7. The traces used are the ones earlier selected and will be displayed one by one. Several options are now possible as can be seen on the menu. Normally no hardcopies are made in single trace mode since it is intended for fast routine work. However, by starting MULPLT in multitrace mode (option 2) and then go to single trace mode (command T(Togg)), hard copy files are made.

Multitrace mode:
In this mode hard copies can be made. If option 2 is used, both screen plot and hard copy files are made. If replot is made, only the last plot is available in the hard copy file. If option 3 is used, which is only hardcopy, there will be additional questions about, window length, start time, scaling and filters. If the scaling is set so that the plot occupies more than one page, several pages will be printed. If in this mode, filenr.lis is given as file name, the program assumes that all the files should be plotted and the only questions will be about the scaling and filters. All channels in each file will be plotted. This option is useful for plotting a large number of events with a single command.

All channel mode
In this mode, all channels for selected stations are displayed in a new window. This mode is particularly useful for working with three component data. By selecting one or several stations in multitrace mode, all components for those stations will be displayed in new window by pressing y or ALLC on menu. Similarly in single trace mode, pressing y or ALLC will display all channels for that station. The user can then go back to e.g. multiTRACE window and select another station to work with in three component mode.

Multiple screens in multitrace mode
If many channels are available (like more than 30), it might be difficult to distinguish all and the channels can be displayed in multiple screens. The number of channels per screen is set in MULPLT.DEF. The number of windows or screens for a particular data set is given in top left hand corner as e.g. ‘Win 2 of 7’ meaning current window is number 2 of 7 windows. To move to the next window, use TAB or NextW in menu. In each window, normal operation can be done. Channels selected will be kept. Using a large data set, the user can then view each window separately, select the channels of interest and when all channels have been viewed, only the selected channels will remain for display. It is possible to togle between showing all channels and multiple screens by pressing N.

Plotting stations in a given distance range When many stations are available, it might be useful to only plot only the stations nearest the epicenter or a particular location. For this option to work, parameter MULPLT AREA in MULPLT.DEF must be set to a value larger than 0.0. For more information, see section Working with many channels in MULPLT.

Channel order in multitrace mode:
Channels are plotted by epicentral distance or in alphabetical order according to station name, see parameter NSORT,DISTANCE and CHANNEL SORTING in MULPLT.DEF. They can also be plotted in the order they are stored in the waveform file(s). By setting the channel order parameter in the MULPLT.DEF file, it is possible to plot the channels in distance or time order. If MULPLT is started from EEV (and NSORT,DISTANCE is set), the channels will be plotted in epicentral distance order unless the station if is not in the station file. If the event does not have an epicenter the location of the station with the first pick will be used as the epicenter for the ordering. Since there is no consideration for channels for the same station, the channels for one station, will be plotted in the same order as given in the waveform file. Ordering can be turned on with the key "-" or pressing (Dist). If set in the MULPLT.DEF file, it is set when MULPLT starts up. It cannot be turned off for a given event when set from MULPLT but the flag is returned to the default value for the next event.

Plotting from continuous data base or archive

If a continuous data base is set up (see section 2.2.3), it is then possible to plot all traces from the continuous data base with MULPLT. When MULPLT starts up, use option cont and the user is prompted for a start time and interval. MULPLT will now check all continuous data bases for available data in required interval and display the available data. The forward (next) or back option will display previous or next window respectively. There is an 25 % overlap between windows. If no data is available for the whole window, no trace is shown. If the beginning and the end is available, a line will join the two segments. If only end or beginning is available, only the available data is shown. All normal operation can be done on the window plotted so it is possible to e.g. extract data. If the register option is used, the whole window is extracted from the continuous data base as one file, copied to WAV and the S-file created.

8.6 Registering new events into SEISAN

Mulpit is the main tool for checking and putting new events into SEISAN. New events with waveform data can appear in two ways in SEISAN:

1. Unprocessed waveform files are available in a work directory and have to be inspected and possibly put into the database. No S-files have been made.

2. Raw data has already been put into a SEISAN database with S-files and corresponding waveform files in some work directory, the data has not been checked. This process has most likely been done with the automatic data collection software SEISNET [Ottemöller and Havskov, 1999], or other similar programs like RTQUAKE (similar to SeisComP and and SC2SEI (gets data form SeisComP), see ftp://ftp.geo.uib.no/pub/seismo/SOFTWARE/. However, events can also have been auto registered with program AUTOREG.

3. Registering from continuous data: SEISAN continuous data base, BUD or SeisComp archives or a large SEED file.

In both cases above, the aim is to inspect an event and decide if the event is real and should be put into the database using option ‘p’. All work must be done from the directory where the raw waveform files are located. The process of putting an event into the database results in creating the S-file (option1), giving the event identifiers and copying the waveform files of registered events to the waveform directory. By pushing p(Regis), the user will be prompted for distance indicator, which has to be L, R or D for local, regional or distant event. It is possible here to enter 2 characters like LE or LV for local explosion or
local volcanic event. The event type or event ID can be any character. Four characters are predefined
and should only be used if the following definition correspond: P(probable explosion), E(explosion),
Q(confirmed earthquake) or V(Volcanic event). If the user enter L, R or D in lower case, the case will
automatically be changed to upper case. The same also happens with E, P, Q and V. A third cahracter
can optionally be entered for the model indicator which is put into column 21 of the header line. The
volcanic events have a sub classification which can be entered when registering an event as volcanic, see
section 39. The process of registering the event into the database implies that a new S-file is created or
registered and in the S-file. An operator ID will be asked and the operator ID will be put on the ID-line.
The question about operator will only be asked for the first event since it is assumed that all subsequent
events are put into the same database by the same operator. The event ID, can later be used with the
SELECT program to select out particular event types. When first putting an event into the database,
the user is also prompted for database.

Option (1)
Data is available as waveform files only and a list of files must be made first with DIRF. Main option
0, 1 or 2 can be used for plotting. The 'p' option creates the S-file and copies the waveform file to the
WAV directory. The waveform file remains in the working directory. Unwanted waveform files can also
be deleted so that when all events have been put in, only waveform files of 'real' events remain in working
directory. These can then be plotted with one run of MULPLT, see section 8.1.

Option (2)
Data is available already in a database, however since the data has not been inspected, the waveform files
are still in a work directory. In EEV, the first unprocessed event in the month is found with command
'ss' and MULPLT is started with command 'po' to invoke all defaults. If the event is to remain in the
database, it must be registered with option 'p'. The process and the questions are the same as in option
(1) except that the S-file is not created since it is already there. The S-file is cleaned for all processing
information from SEISNET if present. This normally also includes automatic phase picks. However, they
can be kept if parameter REG_KEEP_AUTO is set in the SEISAN.DEF file. The status of the files also
changes to being newly registered as under option (1) (see definition of processing codes in Appendix 1)
and waveform file(s) copied to WAV. Before registering, it might be an advantage to merge waveform
files and delete unwanted files (could be false triggers), see section 8.5. Files can only be merged and
deleted in working directory with commands Delw and Merge (Menu). Delw command only deletes the
waveform file names in the S-file. In this process of putting new events into the database, it is also an
advantage to delete unwanted events. This is done with option 'S'(Del S). The S-file is deleted, but the
waveform files remain in the working directory.

Option (3)
When reading continuous data, an output file can be made with the Out option and that output file
can be registered. However, it is also possible to do this in a single operation using the Register option
like above. The selected time window is then extracted, the waveform file copied to WAV and the S-file
registration made. In addition to the questions above, additional questions are given:

Output channels on screen: s
No output, just register: n
No output, just AR line: a
Output all channels: enter

These option are intended to be used with routine operation when many channels are present and the
user only view a few, like all vertical channels, but want to extract (and register a file with all channels).
For ARC archive, the user might want to work directly with the archive from EEV and thus not extract out a waveform file but only create a reference to the archive.

Preprocessing of data while registering new events, option (1)

Normally a series of events are registered first and MULPLT terminated. Then EEV or SE is started up for interactive picking and location. However, if preliminary processing is desired while registering the event, this is also possible.

Phase picks: If phases are picked before the event is registered, these readings are saved in the database at the time of registration. After the event has been registered, MULPLT automatically goes to the next event in FILENR.LIS and no more phase picking can be done.

Processing with a given program: Optionally MULPLT can, after registration, start any program processing the newly registered event. E.g. the AUTOPIc program can be started or a program reading amplitudes etc. The program name is defined in MULPLT.DEF.

Locating the event: As the final step after registration, the event can optionally be located and the location optionally placed in the database.

The above options have been put in on the suggestion of Brian Baptie, who is using it for rapid processing of volcanic events, where in most cases the operator only wants to look at the event once.

8.7 Phase picking, amplitude, weight and polarity

Picking phases:

The plot will display any pick present in the database (current S-file). To pick new phases, position cursor at phase, and press the key as indicated on top of the screen (if in Single mode). E.g. pressing 1 will read IP. Pressing the same key again with the cursor at a different place will delete the old one (indicated with a D) and display the new one. Additional default phases, which can be picked, are i for I ans e for E (note upper or lower case). Keys for phases have default definitions, but can be redefined using the file MULPLT.DEF, see below. The end of the coda is picked as a phase (C) and the program calculates coda length IF AND ONLY IF A P-READING IS PRESENT.

Picking amplitudes:

Position the cursor at the bottom or top of a wave and press a, then at the other extreme (bottom or top) and press a (do not use upper case, see below). There is no requirement for going left to right or top to bottom, it can be done in any order as long as the two extremes are marked. At each press, a cross is marking where the pick was made. In case a filter, like WA, MS or Mb is applied, the program will associate the amplitude with the respective amplitude phase (IAMl, IAMs,20,IVmB,BB, IVMs,BB or IAMb). Amplitude and period are calculated and stored with the phase. Otherwise, if none of these filters are applied, a menu pops up and the user needs to select an amplitude phases name phase name to which the amplitude and period readings are associated. If an attempt is made to pick amplitude on a trace which is not in nm or nm/s, the reading must be confirmed since SEISAN assumes all amplitudes to be in nm or nm/s (see section on instrument correction). If no amplitude phase is picked, no amplitude is stored. The amplitudes are always assumed to come in pairs so if e.g. 3 amplitude values have been picked, and the user tries to pick a phase or quit the program, it will appear frozen since the program is still waiting for the next amplitude measurement. It is always the last pair of amplitude measurements, which are used. Amplitudes can be picked on both corrected and uncorrected traces.
If A is pressed instead of a, the amplitude is read and marked automatically. It works in most cases, but sometimes two subsequent peaks are not correctly chosen and the amplitude reading has to be done manually. The method is to find the absolute extreme and then the largest amplitude before or after is selected in order to obtain the peak to peak amplitude, from which the amplitude is calculated by dividing by 2. For more information, see ../LIB/auto_amp.for.

Manual and automatic amplitude picking can be done in both single trace and multi trace mode. In multi trace mode the amplitude will be picked at the trace nearest the cursor.

Component names when picking phases:
In the old Nordic format S-file (the default), the component only has 2 letters while in the waveform file it has 3 lettersi (in old SEISAN waveform files it can be 4, bu trare used). There must therefore be a unique translation between the two. This definition is given in the subroutine componen.for in LIB. Most common combinations are now defined, however if a new one is defined in the waveform file which does not exist in componen.for, the first and last letter of the input component will be used. If e.g. an input component is called SHZ, then the code in the S-file will be SZ. This means that picks for stations with components, which do not differ in first and last character, cannot be separated in the S-file. Component names for rotated channels will be e.g. SR and ST for short period radial and transverse components respectively.

In the new Nordic format there is room for 3 character component names and in addition network and location codes are stored. So there is no problem with identifying a channel. The complete station, component, network and location codes will be used to identify a reading.

Reading polarity:
If the cursor is above or below the trace at a distance marked by horizontal tics on the sides of the plot, the first motion is also picked and displayed when the next phase is picked. So if e.g. picking a P with dilatation, put cursor below the line and pick P. Do not use a filter if possible. Note that if you delete the phase later, the polarity is also deleted.

Assigning weight:
A phase can be assigned a weight. Move the cursor close to a pick and press one of the keys 0-9 in UPPER case thus using e.g. ! # (default, can also be changed), and a HYPO style weight is assigned and displayed. Although weights 0 to 9 can be put in, HYP only uses 0-4 and 9 (see section 7.1). Phases with associated amplitude, period, azimuth or apparent velocity are displayed with a hat below on the phase indicator line. The default keys for the weights might not be correct on all keyboards, if not, set keys in MULPLT.DEF using the parameter PHASE WEIGHT KEY.

The keys are defined as follows

<table>
<thead>
<tr>
<th>Weight</th>
<th>Linux</th>
<th>Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>!</td>
</tr>
<tr>
<td>2</td>
<td>@</td>
<td>&quot;$&quot;</td>
</tr>
<tr>
<td>3</td>
<td>#</td>
<td>#</td>
</tr>
<tr>
<td>4</td>
<td>$</td>
<td>$</td>
</tr>
<tr>
<td>5</td>
<td>%</td>
<td>%</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>&amp;</td>
</tr>
<tr>
<td>7</td>
<td>&amp;</td>
<td>/</td>
</tr>
<tr>
<td>8</td>
<td>*</td>
<td>(</td>
</tr>
<tr>
<td>9</td>
<td>)</td>
<td>=</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Automatic determination of coda length (C or c):

The coda length can be quite variable among different operators and a function has been made to automatically determine the coda length. The signal is bandpass filtered and the end of the coda is determined by a standard STA/LTA procedure. The parameters are set in the MULPLT.DEF file. Press C to find coda length automatically or c to determine manually. If parameter CODA AUTO is set in MULPLT.DEF, c is used.

In old Nordic format

the position of the coda can only be determined if a P-phase is present since the coda length (time between coda pick and P) is stored together with the P. The location of the pick is indicated with Coda on the plot.

In new Nordic format

the coda can be picked even when there is no P since the coda is now an independent phase called END (IASPEI convention, indicating end of earthquake signal). However, the coda length is only given (on phase line END) if there is a P.

From SEISDAN version 12, coda is no longer stored with the P but with the phase END. However, magnitude is still calculated when coda is stored with the P.

Below is shown an example of old and new format.

Old:

<table>
<thead>
<tr>
<th>STAT</th>
<th>SP</th>
<th>IPHASW</th>
<th>D</th>
<th>HRMM</th>
<th>SECON</th>
<th>CODA</th>
<th>AMPLIT</th>
<th>PERI</th>
<th>AZIMU</th>
<th>VELO</th>
<th>AIN</th>
<th>AR</th>
<th>TRES</th>
<th>W</th>
<th>DIS</th>
<th>CAZ7</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASK</td>
<td>SZ</td>
<td>IP</td>
<td></td>
<td>337</td>
<td>56.25</td>
<td>88</td>
<td>62</td>
<td>0.38010</td>
<td>171</td>
<td>140</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

New:

<table>
<thead>
<tr>
<th>STAT</th>
<th>COM</th>
<th>NTLO</th>
<th>IPHASE</th>
<th>W</th>
<th>HHMM</th>
<th>SS.SSS</th>
<th>PAR1</th>
<th>PAR2</th>
<th>AGA</th>
<th>OPE</th>
<th>AIN</th>
<th>RES</th>
<th>W</th>
<th>DIS</th>
<th>CAZ7</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASK</td>
<td>S</td>
<td>Z</td>
<td>IP</td>
<td></td>
<td>337</td>
<td>56.250</td>
<td>RRR</td>
<td>dd</td>
<td>62.00.38010</td>
<td>171</td>
<td>140</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASK</td>
<td>S</td>
<td>Z</td>
<td>END</td>
<td></td>
<td>339</td>
<td>24.680</td>
<td>88.0</td>
<td>171</td>
<td>140</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\section{Theoretical arrival times for global and local phases and location}

In order to assist in identifying seismic phases, there is an option for displaying the theoretical travel times for the current event while picking phases (using ttlocal based on layered model in STATION0.HYP). The steps to do so are the following:

\begin{itemize}
\item Before entering MULPLT from EEV, the theoretical travel times can be calculated for the current event. This assumes that the S-file has many phases. The times will be calculated for all stations in waveform file(s). In both cases, the times are stored in file iasp.out (no importance for the user, just for information). See Figure \ref{fig:mulplt-multitrace} for an example in multitrace mode. Note that very many theoretical phases can be generated if the S-file has many phases.
\end{itemize}
When a trace is displayed on the screen, all theoretical phases inside the time window will be shown. To distinguish the theoretical phases, they are prefixed with a y and displayed below the trace (normal phases have I, E or blank and are displayed above the trace). Position cursor where you see a phase which you think corresponds to a theoretical phase and press \texttt{y}. The nearest theoretical phase will now be placed at that position with a prefix E. Only theoretical phases selected in this way will be written in the S-file. Note that phase names can be up to 8 characters long, see Appendix 1 for the definition of long phase names when using old Nordic format.

If the phases fit badly, start looking at the P-phase. If that does not fit the theoretical P-phase, change the origin time in the S-file so that the P-arrival fits, and recalculate the theoretical phases.

**PROBLEM:** In multitrace mode, only one theoretical phase can be picked. Replot must be made before picking the next.

Locate earthquake

**Instrument correction and magnitudes Ml, mb and Ms**

The correction for instrument response is done by taking the spectrum of the selected window of the trace, dividing with the spectrum of the theoretical phase of the same frequency, and then multiplying with the inverse of that value. Filtering is done in the time domain by first applying a sine taper on 10 percent of the signal in each end, then doing the FFT. If data has been high pass filtered in time domain at frequency \(f\), then a frequency domain 4 pole low pass Butterworth filter is added at frequency \(f/5\). This is for stability.

Correct for response

Do inverse FFT

Ground motion

Option g (Ground) removes the effect of the instrument and displays a ground motion seismogram. Amplitudes for determining Ml (also Mn)

Amplitudes for Ml are normally picked from the S/Lg wave, either vertical or horizontal channels depending how the magnitude scale is defined. For Mn, the amplitudes are read from...
either Pn or Sn, which ever is the largest.

Displaying response information

The response function for the current channel can be shown with option ‘:’ (Resp), see Figure \ref{fig:mulplt-resp}. If no response function is given, a message is shown. If the response function is taken from the waveform file header, amplitude values can only be done with very old waveform files in SEISAN format. \index{Response, from where} \index{Response, show curves}

Amplitude for determining \index{mb}mb:

Determining mb assumes that the maximum amplitudes are measured on classical 1 Hz WWSSN instruments having a peak gain of 60. In SEISAN version to 8.2.1, the default filters used to simulate SP WWSSN were, by mistake, in the band 0.9 (2 pole) to 2 Hz. The Butterworth filter 0.033 - 5 Hz, 8 poles, is hardwired and it cannot be modified with additional filters.

Amplitude for determining \index{Ms}Ms:

The attenuation function for determining Ms assumes that the amplitudes are measured on classical LP WWSSN instruments having a peak gain of 6. For SEISAN 8.2.1, the default filters used were in the band 0.038 (2 pole) to 0.1 Hz (1 pole). Prior to SEISAN 8.2.1, the filters were in the band 0.038 to 0.17 Hz with 2 poles. These changes might have resulted in small errors in Ms and can only be corrected by repicking the amplitudes.

Amplitude for determining \index{Ms}Ms:

Amplitude for MB is defined as the maximum velocity on a wide band instrument (0.2 -30 sec or 0.03 -5 Hz). The Butterworth filter 0.033 - 5 Hz, 8 poles, is hardwired and it cannot be modified with additional filters.

Amplitude for determining \index{Ms}Ms:

The attenuation function for determining Ms assumes that the amplitudes are measured in the period range 18 - 22 sec and it is up to the user to make sure that the amplitude is in the correct range.

For SEISAN 8.2.1, the default filters used were in the band 0.038 (2 pole) to 0.1 Hz (1 pole). Prior to SEISAN 8.2.1, the filters were in the band 0.038 to 0.17 Hz with 2 poles. These changes might have resulted in small errors in Ms and can only be corrected by repicking the amplitudes.

Amplitude for determining \index{Ms}Ms:

Amplitude for MS is defined as the maximum velocity on a wide band instrument (3 -60 sec or 0.017 - 0.3 Hz). Using the Butterworth filter 0.017 - 0.3 Hz, 8 poles, is hardwired and it cannot be modified with additional filters.

Problem: If a long trace (large number of samples) is used, the instrument correction might fail (funny result seen) due to numerical overflow in the spectral conversion. Choose a shorter window. \index{Problem, instrument correction}

\section{Determine azimuth of arrival (3 comp or array) and component rotation}

Azimuth of arrival (back azimuth) from 3-component stations, h(Azim) if a 3 component station is available, the azimuth of arrival can be determined using the method developed by \cite{roberts1989}. Display any of the 3 components and press h (Azim). Then select a zoom window around the P-arrival of a few secs duration for the analysis. The 3 components will now be displayed below in order Z, N and E and the calculated azimuth, apparent velocity and correlation will be displayed at the bottom line. In order to check the stability of the estimate, try different windows and filters. Often, a filter must be used to get reliable results. The displayed azimuth and apparent velocity is only saved in the S-file when an associated phase is picked. THAT PHASE MUST BE PICKED ON THE SINGLE UPPER
TRACE SEEN ON THE SAME SCREEN. If there is none, use I or E. The velocity estimate is not very reliable and is dependent on the local velocities. In order to calculate the apparent velocity, the P-velocity of the top layer must be given. The default value is 5.0 km/sec, but another value can be set in the MULPLT.DEF file. To get a good estimate, the correlation coefficient should be as high as possible and positive. The quality of the obtained azimuth can be tested by locating the event with the calculated azimuth weighted out and observe the azimuth residual.

The output is different in new and old Nordic format, see example below:

\verbatim
old:
STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZimu Velo AIN AR TRES W DIS CAZ7
ASK EZ EP 916 40.94 37.3 9.2 111 29.9 205
ASK EZ E 916 41.96 41.8 7.8

new:
STAT COM NTLO IPHAS W HHMM SS.SSS PAR1 PAR2 AGA OPE AIN RES W DIS CAZ7
ASK EHZ EP 916 40.940 111.0-1.0510 29.9 205
ASK EHZ BAZ-P 916 40.942 37.3 9.3 BER jh
ASK EHZ BAZ 916 41.962 41.8 7.8 BER jh
\endverbatim

Two back azimuths (BAZ) have been picked, one associated with the P and one a bit after the P. In the old format, the one picked with the P is also stored with the P-phase line since there is space on every phase line for BAZ while the one picked later is put on a new phase line and marked with E.

In the new format the 3 observations are on separate lines and the phase picked with the P is identified as BAZ-P and the BAZ pick with E is marked as BAZ and no phasse. The advantage of 3 lines is that each observation can be changed independently of the others.

Figure 8.10 shows an example.

Azimuth and apparent velocity from array data, FK analysis f(FK)

Using this command, the traces seen on the screen will be put into the FK program and an FK plot will be displayed. The azimuth and apparent velocity with the highest correlation is selected by the program, however any other value can be manually selected. The values will ONLY enter the S-file if associated with a phase in the same way as amplitudes are picked. For more details, see section 40.

Rotated seismograms

Option U(Rotat) will rotate the horizontal components for the next plot if the two horizontal components are available. The rotation will display the radial component instead of the N-component and the transverse component instead of the E-component. The back-azimuth used is displayed above the trace. All channels will be displayed rotated until u(Rotat) is pressed again. This means that phases can be picked and spectra made with the rotated channel. When picking phases on rotated signals, these will appear in the S-file with components R or T instead of N and E respectively. This also means that only if the rotated signals are shown, will the phases read on rotated channels appear on the plot. The station
back-azimuth is obtained in the following way: If a hypocenter is given in the header line, the angles are calculated using the current $\text{STATIONx.HYP}$ file. If no hypocenter is available, the angle will be read from the S-file under column observed azimuth (47-51) (if not blank) and the azimuth residual will be added. This option permits the user to first determine the azimuth with the 3-component option and then rotate the signals with the determined azimuth. Finally, if no observed azimuth is available, the event to station azimuth + 180 deg. will be used if available (column 77-79). If no back-azimuth can be found, no rotation is done and an angle of 999 deg. is displayed. If in single trace mode and choosing the 3-component option AND the rotate option, the user will be prompted for a rotation angle and the rotated channels will be shown in the usual 3-component plot, however, the azimuth determined is done with the unrotated channels.

PROBLEM: In general, the R-channel will use the response of the N-channel and the T-channel will use the response of the E-channel so for instrument response removal to be correct, the 2 channels must have the same response curve.

### 8.8 Data manipulation commands

Select other channels: o(Oth)

The channel selection menu comes up again.

Go back one channel in single trace mode, go back one event in multitrace mode if MULPLT is started from EEV: B(Back)

Select other waveform files from S-file: W(OthW)

If more than one waveform file available for the event, one or several others can be selected.

Delete waveform files:

This can only be done in multitrace mode: The command is d(DelW) and the cursor must be above the top frame of the plot. There are two possibilities:

1. Input is from `filenr.lis`: The current file is deleted and if in default mode, the plot moves on to the next event.

2. MULPLT is started from EEV: If only one waveform file is available, the program proceeds as under (1). The waveform file is deleted and the waveform file entry in the S-file remains. However, if more than one waveform file is available, the user can use a menu to select which files to delete. Only the waveform file entries in the S-file are deleted, the waveform files remain. This option is mostly used with SEISNET.

Delete S-files D(Del S)

This command deletes the current S-file. It can only be used if MULPLT is called from EEV. No waveform files are deleted.

Merge waveform files given in S-file M(Merge)

The files will be merged to one waveform file and the old individual file names removed from the S-file and replaced by the new file name of the merged file. The original waveform files remain. Files to be merged will be shown on a menu. Mostly used with SEISNET. The user MUST have files in working directory. If files are in the data base, they will not be shown on the merge menu.
Overlay two channels: It is sometimes practical to be able to overlay 2 or more channels. The channels to be overlaid must follow each other on the screen (sorting might influence that). Move the cursor to the channel name of the lower of the two channels, press & and the channel is marked with a cross. When doing replot, that channel will be plotted in red on top of channel above. Both channels will be autoscaled, but if absolute scaling is used, the real difference in amplitude is seen. Overlay cannot be deselected without leaving MULPLT. This option is particularly useful when comparing real and synthetic seismograms results from moment tensor inversion, see that section later.

Output of binary waveform file, O(Out)

It is often useful to be able to select part of a waveform file and save it. The Out option makes an output file of the traces AS DISPLAYED ON THE SCREEN with exactly the same channels, and time window in a file with a standard SEISAN waveform name. The output format is by default SEISAN, even if some input files have a different format. A different output format can be selected in MULPLT.DEF, parameter MULPLT WAV OUT FORMAT. The network code in the file name will ALWAYS be the station code if all channels are from the same station. Otherwise the network code has the default name MERGE. Alternatively the parameter MERGE_WAVEFORM can be set in SEISAN.DEF. The data is output exactly as displayed on the last screen, so if filtering or instrument response has been made, the output file will also be filtered or instrument corrected WITH SOME RESTRICTIONS: Not all response, channel or filter combinations are possible. Only 4 pole band pass filters can be used, no rotated channels and none of the magnitude simulated traces like Wood Anderson for MI. If you want exactly what is seen on the screen for all combinations, use option OutW. If any filtering or instrument response correction has been done, a note will be inserted in the SEISAN waveform header so the user can see that this is no longer the original data. The note could be e.g. ‘Displacement 1.250-20.0 Hz’ indicating that output has been filtered and converted to displacement (nm). Note that numbers have been scaled so only if the SEISAN file is read with standard SEISAN routines, will the numbers be correct. If the output file is converted to ASCII by SEIASC, the number shown must be multiplied with a given scaling factor, see SEISAN binary format description (Appendix B). There is no response information in the header other than the short text. Since the station code is still the same, it is technically possible to correct for the response again using the response information in the CAL directory, however, be aware that this will give wrong results.

Option OutW: This option will output the signal exactly as seen on the screen with all the selected filter and response combinations. It also handles rotated channels. The output file is mulplt.wav and it is an Ascii file with real numbers in Helmberger format (readable by SEISAN). To make the file, press OutW and wait for message ‘mulplt.wav written’ in top right of the screen. For many channels and high sample rate it could take some time since the output is Ascii. If the traces in the window do not start and stop at the ends, then dc levels will be added so all channels in the output file has same start and stop times. Format description is found in section moment tensor inversion.

Output of ASCII waveform file

This option only works if parameter SPECTRAL_OUTPUT has been set in MULPLT.DEF. The output file signal.out contains the last data displayed in the single trace zoom window (in ASCII and real numbers). This option is a another way (see option O(Out) above) of getting an output file that has been filtered or instrument response corrected. The main difference is that this file is only for one trace written in ASCII.

Fixed scaling

Normally all traces are plotted with autoscaling. However, it is sometimes useful to be able to scale the traces with a fixed scale in order to e.g. compare traces or override the autoscale in case a spike distorts the autoscaling. Option *(Scale) will prompt the user for a maximum count to use for the scaling of all
traces.

Example of using MULPLT on Linux:

Comments are given with ! in front

This example shows how running MULPLT from EEV would look.

```
/top/seismo/REA/BER__/1991/01/01 0557 12L.S199101 ! S-file name
Read header from file /top/seismo/WAV/9101-01-0557-12.WNN_13
Plot options: Interactive picking Return ! first choice
  Multi trace plot on screen, def (0)
  Multi trace plot on screen (1)
  Multi trace plot on screen+laser(2)
  Multi trace plot on laser (3)
  Multi trace plot on laser (3)
  Continuous on screen (4)
  Continuous on screen + laser (5)
  Continuous on laser (6)
  Stop (9)

! now comes a menu for selection and then
! the plot appear in Single mode since a
! return was made
```

The next example shows how to plot many events in one go, first make a list with DIRF.
8.8. DATA MANIPULATION COMMANDS

Figure 8.7: Using MULPLT for picking phases. The top shows the original trace and the bottom the zoomed part. Note that the amplitude has been associated with the phase E and not the ESg. This means that if the S-phase is deleted, the amplitude will remain. In the new Nordic format, amplitudes are always stored on separate phase lines.
```
dirf 9101-10*  ! events from January 10, 1991
# 1 9101-10-0915-15S.KMY_03
# 2 9101-10-1510-55S.NSS_12
# 3 9101-10-2333-44S.NNN_11

mulplt
file name, number, filenr.lis for all
filenr.lis  ! plot all events in filenr.lis
Resolution in cm/sec, 0: plot all on one page (default)
0  ! scale will be different for each plot!!!
Read header from file:9101-10-0915-15S.KMY_03
Page
Channel: 1

Pltfile sent
Read header from file:9101-10-1510-55S.NSS_12  ! next event in list
Page
Channel: 1
Channel: 2
Channel: 3
Channel: 4
Channel: 5
Channel: 6
Channel: 7
Channel: 8
Channel: 9
Channel: 10
Channel: 11
Channel: 12

Read header from file:9101-10-2333-44S.NNN_11
Page
Channel: 1

! etc.

Pltfile sent
Read header from file:9101-10-1510-55S.NSS_12  ! next event in list
Page
Channel: 1
Channel: 2
Channel: 3
Channel: 4
Channel: 5
Channel: 6
Channel: 7
Channel: 8
Channel: 9
Channel: 10
Channel: 11
Channel: 12

Read header from file:9101-10-2333-44S.NNN_11
Page
Channel: 1

! etc.
```
8.9 Spectral analysis, s(Spec)

The spectral analysis option for local and teleseismic events is selected in single trace mode. The spectral analysis is based on the Brune [1970] model and various assumptions about the geometrical spreading and anelastic attenuation. The steps in the analysis is:

- Remove DC
- Apply sine taper on 10 percent of the signal in each end
- Do FFT
- Do attenuation correction
- Correct for response

The theoretical displacement spectrum $d(f)$ [Brune, 1970] is:

$$d(f) = G(r, h) \times D(f) \times \text{Moment} \times KK/(1 + f^2/f_0^2) \times (4 \times \pi \times DE \times V^3))$$

where $G(r, h)$ is geometrical spreading, $r$ is epicentral distance, $h$ is hypocentral depth, $D(f)$ the diminution function due to anelastic attenuation, $f$ is the frequency, $DE$ the density, $V$ the velocity at the source, $f_0$ the corner frequency and $KK$ a factor of $2.0 \times 0.6$ to correct for the free surface effect and radiation pattern.

The diminution function $D(f)$ is written as

$$D(f) = P(f) \times \exp(-\pi \times f \times \text{trtime}/(q0 \times f \times \text{qalpha}))$$

where $\text{trtime}$ is the travel time from the origin time to the start of the spectral window and

$$P(f) = \exp(-\pi \times \text{kappa} \times f)$$

is meant to account for near surface losses [Singh et al., 1982] with the constant kappa having a value of the order 0.02 sec. Anelastic attenuation $Q$ is assumed to be frequency dependent following the relation $Q = q0 \times f \times \text{qalpha}$. For $f$ less than 1 Hz, $Q$ is, by default, assumed frequency independent with the value $q0$ (before Dec 2013 it was frequency dependent for $f$ less than 1 Hz which could lead to a small overestimation of seismic moments for events larger than 4.5. This change is also affects other programs using Q-correction like SPEC, AUTOMAG and AUTOSIG). The transition of the Q from high to low frequencies as a function of frequency is made with the function $Q=Q0*(1+f/X)^{qalpha}$ with default parameter $X=1.0$. The parameter $X$ can have other values than 1.0 if the spectral model is used, see below. So $X$ is the transition frequency between frequency dependent Q and frequency independent Q. This parameter is set in SEISAN.DEF. For teleseismic events, only $t^*$ is used and $Q$ must be set to zero (not used). The $t^*$ parameter is the same as kappa and is usually set to 1.0 (same value is used for P and S).

The geometrical spreading has been defined to be dependent on the wave type with several possibilities, all made equivalent to a distance called geo_distance (GD) such that geometrical spreading is expressed as $1/GD$. There are several possibilities for GD:

- **Local and regional events geometrical spreading**
  - P-waves:
    - GD is the hypocentral distance ($HD$) = $\sqrt{(r \times r + h \times h)}$ so body wave spreading is assumed.
S-waves:
The geometrical spreading has been made dependent on distance and depth. At short distances, the geometrical spreading is assumed to be body wave spreading. For distances beyond the Herrmann-Kijko distance (default of 100 km) and a shallow focus, the following relation is used:

\[ G(r, h) = \frac{1}{r} = \frac{1}{GD} \text{ for } r < 100 \text{ km} \]

\[ G(r, h) = \frac{1}{\sqrt{100 \times r}} = \frac{1}{GD} \text{ for } r > 100 \text{ km} \]

which is commonly used \cite{Herrmann1985, Herrmann1983}. This relation assumes surface wave dispersion for epicentral distances larger than 100 km. In SEISAN, 100 km is the default, however it can also be set to any other value by the parameter HERKIJ_DISTANCE (see later).

The above relation breaks down if the depth is large or comparable to the epicentral distance and in that case body wave spreading is again assumed. In order to get a smooth transition from surface wave to body wave spreading, it is assumed that the relation changes nearly linearly from surface wave spreading to body wave spreading between the depths GEO_DEPTH1 to GEO_DEPTH2. For depths less than GEO_DEPTH1 (default 50 km), Herrmann-Kijko spreading is assumed, for depths larger than GEO_DEPTH2 (default 100 km), body wave spreading is assumed with the transition in between. In each case the geometrical spreading term is given as the equivalent GD, which is also recorded in the database. These 3 parameters can be used to change geometrical spreading. If e.g. HERKIJ_DISTANCE is 10 000 km, body wave spreading is always used. For more info, see \cite{Havskov2010}.

Geometrical spreading for teleseismic events

The geometrical spreading is approximated with \cite{Havskov2010}

\[ G(r) = \frac{1}{GD} \text{ where } GD = \frac{(27 + \text{dist})}{0.0048} \]

where dist is epicentral distance in degrees. This approximation is only valid for depth is less then 100 km and dist larger than 30 degrees.

Attenuation and velocity specification for spectral analysis

There are 2 possibilities, use the values in MULPLT.DEF or use a spectral model in SEISAN.DEF.

Use the values given in MULPLT.DEF: Since only one value can be given for attenuation (Q and kappa), the same attenuation is used for P and S. Likewise, if the hypocenter changes depth, it will be the same fixed values for all attenuation and velocity parameters. For a particular fixed situation e.g. where the user always analyze shallow events and always is using the same type of spectrum (usually S), this is adequate. However in a situation with large depth variations, it is desirable to change all parameters as a function of depth. It might also be useful to change attenuation as a function of phase type although there often is little difference between P and S attenuation. The value of Q is also fixed to be constant below 1 Hz.

Use values in SEISAN.DEF: The spectral model is defined in SEISAN.DEF. In order to activate the model, set parameter SPECTRAL MODEL in MULPLT.DEF to 1.0, then the spectral model is used instead of the parameters in MULPLT.DEF. The model gives velocities (P and S), attenuation (P and S) at different depths as will as kappa for P and kappa for S and the behavior of Q below 1Hz (P and S). Remember that the attenuation given at a particular depth is NOT the attenuation at that depth but the average between source and receiver. Values used will be interpolated between values in model and the values used will be show in the plot and stored in the S-file. If the spec model is used, it will be
indicated on the plot (lower right hand corner). If a spec model is not there, MULPLT.DEF values will be used and also indicated on the plot.

From the spectral parameters, source radius and stress drop can be calculated as follows:

\[
\text{Source radius} = 0.37 \times \frac{V}{f_0}
\]

where \(f_0\) is the corner frequency and \(V\) the P or S-velocity at the source for P and S-spectra, respectively.

\[
\text{Stressdrop} = 0.44 \times \frac{\text{Moment}}{(\text{source radius})^3}
\]

The spectral analysis is used in two ways. The first and most common is to make the attenuation and instrument corrected displacement spectrum and determine the flat spectral level \(OM_0\), and corner frequency \(f_0\) from which the seismic moment, source radius and stress drop can be calculated. The second option is to display the instrument corrected spectrum (displacement, velocity or acceleration) and model the spectrum for corner frequency and attenuation parameters. In this case no correction for attenuation should be made.

Spectral analysis to determine moment, source radius and stress drop:
Select the spectral option, s(Spec). Before the spectrum comes up, you will get a question of the type of spectrum wanted and a few other options.

Displacement: d Displacement spectrum
Velocity: v Velocity spectrum
Acceleration: a Acceleration spectrum
Raw spectrum: r Spectrum without instrument correction
Change velocity: c Use other velocity for calculation
Change density: e Use other density for calculation
Change moment: m Change moment (see modeling)
Noise Pow. spec: n Noise power spectrum (see later)
Lin axis: l Change to linear axis
New spec f-lim: f Change frequency limits for spectrum
Change spectrum: t Change to P or S spectrum if auto does not work
Autofit spectrum: s Make automatic spec fit

The most used possibilities are displacement (d), velocity (v) or acceleration (a). For determination of Moment etc, the displacement spectrum or the autofit option (see below) MUST be selected. Unless raw spectrum is selected, the spectrum will be instrument corrected. If no response file is available in CAL, a message will be displayed on the screen and the raw spectrum calculated. At this stage it is also possible to change the velocity from the MULPLT.DEF value or the moment to be used for spectral fitting as given in the S-file (see spectral fitting below). The velocity change is only affective for the current spectrum since the user might make both P and S spectra and it is up to the user to make sure that it is appropriate for P or S, however, the value is saved in the S-file. The density change is kept for the whole MULPLT session since there is no change with velocity and its value is also saved in the S-file. The spectrum shown will normally show both the spectrum from the selected time window as well as a noise spectrum from an identical length time window at the start of the trace. IF NO NOISE SPECTRUM is desired, select spectrum with capital S instead of s.

Option s, auto spectrum
If auto spectrum is chosen (s), an attempt will be made to fit the Brune spectrum to the observed spectrum. The program will first try to find the frequency range of acceptable S/N and then find the
best fit by grid search and the fit will be shown on the spectrum as well as the frequency range used. The automatic frequency range determination can fail, particularly for small events so a check should be made if the spectrum fit looks ‘reasonable’ and that the frequency range is ok. Choosing a slightly different window might fix the problem. The initial frequency range is limited by the parameter SPECTRAL F-BAND set in MULPLT.DEF so changing this parameter might also help. Note that the automatic frequency range determination will fail if the noise window before the P is not as long as the analysis window chosen. Automatic spectra can also be done for the whole event with command automag in EEV, AUTOMAG and AUTO from the prompt line.

The spectral analysis produces two output files:

com_spec.out: The complex spectrum with some additional information needed for surface wave analysis, must be displacement spectrum.

amp_spec.out : The real spectrum given as frequencies and amplitudes. The files are only generated if parameter SPECTRAL_OUTPUT is set in MULPLT.DEF. Setting this parameter will also generate an ASCII waveform file with the input signal used.

Power spectra: The above spectra can also be displayed as power spectra if capital letters are used. Using e.g. ‘V’ instead of ‘v’ will show the power velocity spectrum.

When the spectrum comes up (see example in Figure 8.9, the axis units are log amplitude in nanometers-sec (displacement) versus log frequency (Hz). The cursor can be used to select the level, corner frequency and slope by defining the spectrum with a 3 point selection. This 3-point selection is finished with f, q or r with the same meaning as in picking mode. The spectral values are displayed on the screen once q, f or r is pressed. The abbreviations are

General parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vel:</td>
<td>Velocity used (km/sec) (Vp or Vs)</td>
</tr>
<tr>
<td>Dens:</td>
<td>Density (g/(cm**3))</td>
</tr>
<tr>
<td>Dist:</td>
<td>Hypocentral distance (km)</td>
</tr>
<tr>
<td>Depth:</td>
<td>Hypocentral depth (km)</td>
</tr>
<tr>
<td>q0:</td>
<td>q0 for spectral amplitude correction</td>
</tr>
<tr>
<td>qalpha:</td>
<td>qalpha for spectral amplitude correction</td>
</tr>
<tr>
<td>k:</td>
<td>kappa</td>
</tr>
<tr>
<td>q1Hz:</td>
<td>frequency where Q(f) changes to constant Q</td>
</tr>
<tr>
<td>spec model:</td>
<td>Spec model is used</td>
</tr>
</tbody>
</table>

NOTE: The velocity is the velocity at the source, so if the dataset contains both shallow and deep earthquakes, a single velocity will be an approximation if the spec model is not used. There are two solutions to this problem if the values in MULPLT.DEF are used. A: Use different MULPLT.DEF for deep and shallow events. Since the the MULPLT.DEF used is taken from working directory if available, you can have different directories when working with different depth earthquakes. B: Use the general MULPLT.DEF in DAT and change the velocity after making the spectrum in the S-file. At the next update, the moment etc will be recalculated with the new velocity. On top of the general parameter is indicated which kind of spectrum is assumed, P or S. In order for the program to automatically determine which kind of spectrum to assume, there must be a P or S reading displayed on the screen near the time window analyzed. The reading must be within 10 sec of the start of the window. If both a P and S-reading is within 10 secs, the nearest phase is chosen. If it cannot be determined which kind of phase is analyzed, the user will get a question to select type of phase (can also be changed later when spectral choices come up) The determination of which phase influences the further calculation of geometrical spreading and moment (uses P or S-velocity).
If f is selected, the spectral values together with calculated moment etc are stored in the S-file at the next key press (see parameters below). Spectral values in S-files accumulate, since no old values are deleted !!! This is because the spectrum might be made under different conditions (start time, time window etc). The input parameters for the spectral analysis is given in file MULPLT.DEF, which can be in either DAT or the working directory, see below. Additional parameters for geometrical spreading are given in SEISAN.DEF in DAT.

The spectral parameters are calculated using the relations

\[ \text{Moment} = 4 \times \pi \times DE \times V \times 3 \times 10 \times OM / (G(r, h) \times KK) \]

where V is the seismic wave velocity at the source (P or S if P or S-spectrum respectively) and OM the spectral flat level on the attenuation corrected displacement spectrum.

\[ \text{Momentmagnitude} = \frac{2}{3} \times \log_{10}(\text{moment}) - 6.06 \]

which is equivalent to the relation

\[ \text{Momentmagnitude} = \frac{2}{3} \times \log_{10}(\text{moment}) - 10.73 \]

if moment is in dynes-cm [Kanamori, 1977].

The moment is calculated in Nm, the source radius in km and the stress drop in bars. All results are written to the S-file. Below is an example:

```
SPEC ITK S Z MO 13.0 ST 4.2 OM 1.5 f0 9.45 R .22 AL 2.50 WI 4.0 MW 2.6 3
SPEC ITK S Z K 0.002 T 7 GD 52 VP 6.00 DE 3.00 QO .0 QA 1.00 Q1 0.0 3
```

Note that no special line has been created in the Nordic format. Comment lines are used with SPEC at the start of the line followed by station and component. Only the first 4 characters of the 5 character station name is used. The station and component names are given at the start of the line. In case of a 5 character station name, the station name is shifted one character to the left. The information is:
MO: log of moment, unit Newton*m
ST: Stress drop in bars
OM: log spectral level (nm-sec) not distance corrected
F0: Corner frequency (Hz)
R : Source radius (km)
AL: Decay of log spectrum
WI: Spectral window used (secs)
MW: Moment magnitude
T : Start time of window for spectrum in hr, min, sec
K: Kappa
GD: GEO_DISTANCE in km
VP or VS: Velocity in km/sec at source for P and S-spectra respectively. The P or S in this line indicated if the spectrum is a P-spectrum or an S-spectrum. It MUST be P or S to be used for magnitude determination. A '?' is put in if MULPLT does not know which kind of spectrum is (no P or S reading near start of spectral window). This can be changed by editing the S-file afterwards.
DE: Density in g/cm**3
Q0: q0 in relation Q = q0 * f ** qalpha
QA: qalpha
Q1: frequency where Q(f) changes to constant Q

Note: Before version 10.1, VS was written instead of Q1. However, VS was not used for anything. Note: The component codes have not been adjusted for SEED so the location code is not included.

Note: In earlier versions (before version 7.0), the field for kappa was used for the travel time to start of window. This can be calculated from origin time and the start time of the window.

NOTE: MOMENT IS NOT CALCULATED IF THE SPECTRUM IS NOT IN DISPLACEMENT.

When doing an UPDATE of the database or just a location with HYP, all distance dependent spectral values are recalculated and average values written into the output file. Mw will be calculated from the average value and written in the header line. However, the original distance dependent Q and kappa correction is not changed, since this correction was used to modify the spectrum used for reading parameters. Normally a small distance change has insignificant influence on the spectral level or the corner frequency so the Q-correction should be no problem. Spectra of the same type (P, S or ?) and from the same channel are overwritten. Only in case of UPDATE are the values written back into the database.

Display of spectral parameters: Program MAG can read and plot relations between spectral and source parameters. Program REPORT can read spectral parameters and combine in a table.

Potential problem with Q-correction: If the origin time in header is wrong, the Q-correction can be very wrong.

There must be a phase line in the S-file with component and distance corresponding to the spectra made in order for the spectral values to be calculated.

Spectral fitting

Once the spectrum has been shown (displacement, velocity or acceleration), a theoretical spectrum can be calculated and superimposed on the observed spectrum in order to forward model either source parameters or attenuation.
8.9. **SPECTRAL ANALYSIS, S(SPEC)**

Another option (s: Plot stored fitted spectrum) if spectral analysis already has been done, is to plot the corresponding theoretical spectrum on top of the calculated spectrum. This option has the purpose to be able to check previously made spectra whether manual or automatic. The SPEC lines have all the information needed to make the theoretical spectrum.

Entering constants and modeling: The modeling can only take place when the spectrum is seen on the screen.

Press S and a question will appear to enter the constants f0, k, Q0 and qα which are as defined above except qα is Qα. Once these parameters have been entered (terminate with return), the theoretical spectrum (displacement, velocity or acceleration depending on what is used for the spectrum) is calculated and superimposed on the observed spectrum. The parameters used or calculated are displayed. S or s can now be pressed again and a new theoretical spectrum calculated and plotted. To get out of the spectral fitting loop, type r or q as usual.

Which constants and parameters are used: The moment is taken from the last moment calculation made for this station, if any. If none, the moment is taken from the S-file if an average moment has been calculated (see UPDATE command). If no moment is available, it can also be entered the first time the spectrum is shown (option M). If no moment is available, no modelling can be done and a message is given. The distance and depth is likewise taken from the S-file. If no distance is available, no modelling can be done and a message is given. If all 4 parameters f0, k, Q, qα are entered, stress drop is calculated with the relation given above. If the corner frequency is given as zero, the user will be asked to enter the stress drop and the corner frequency is calculated from the stress drop. If Q is zero, no Q-correction is made. IMPORTANT: The Q and qα used here are distinct from the Q0 and Qα used for making the amplitude spectrum and both should not be used when modeling since this would imply a Q-correction two times. The best way is to use Q0=0 and kappa=0, so that Q is only corrected for when modeling. The distance used is everywhere is GEO DISTANCE.

The spectral parameters shown are:

| Moment: | Moment used |
| Geo dist: | Geo distance used |
| Stress drop: | Stress drop in bars |
| f0: | Corner frequency |
| q: | q0 used in spectral fitting |
| qa: | qalpha used in spectral fitting |

**Power spectrum and noise spectrum**

The 3 types of spectra (displacement, velocity and acceleration) can optionally be made as power spectra. Instead of selecting the type of spectrum by pressing d, v or a, just press the same characters in upper case and the power spectrum will be shown.

In seismic noise studies, the seismic background noise is often displayed as acceleration power spectral density in dB relative to ((1m/s**2)**2)/Hz. Instead of selecting d, v or a, press n instead. The plot shows the Peterson [1993] new global high and low noise models superimposed on the observed spectrum (Figure 8.8). When doing noise spectra, no attenuation correction is done. The normalization of the spectrum is as follows

\[ P = |F^{DFT}|^2 \times \frac{\Delta t^2}{T} \times 2 \]
where \( P \) is the Peterson Power spectrum, \( F^{DFT} \) is the discrete Fourier transform, \( \Delta t \) is the sample interval and \( T \) is the length of the time window. The factor 2 comes from the fact that only the positive frequencies are used so only half the energy is accounted for. The total power is proportional to the length of the time window since the noise is considered stationary, so by normalizing by \( T \), the length of the time window should not influence the results. This noise option is a handy method of checking the noise characteristics of a given seismic station and compare it to global standards. This kind of analysis can also be done with the SPEC program (section 33). For more information, see instrument.pdf in INF.

**Problems:** There is currently no check if a displacement seismogram has been calculated when calculating the spectral parameters. If spectral analysis is done outside EEV (output in MULPLT.OUT) or with EEV when there is no origin time and/or epicentral distance, the output results are wrong for moment etc. Before calculating moment etc, the S-file MUST HAVE BEEN UPDATED SINCE BOTH THE DISTANCE AND ORIGIN TIMES ARE USED. If the spectra get very high amplitude levels when correcting for instrument, this might be caused by correcting for Q. With a Q of 100 and a distance of 10 000 km, this gives a very large correction. The Q-correction can be disabled in the MULPLT.DEF file. If picks are made, but no readings appear in the S-file or readings appear with wrong component, the waveform file component might not have been defined in subroutine componen.for. If poles and zeros are used to remove the response, rotation cannot be used at the same time.

### 8.10 Spectrogram

SEISAN has two options for making spectrograms. The first is the built in SEISAN routines in MULPLT and the second is an ObsPy script. The first can make a spectrogram of all the traces seen in the multi trace window, while the second make of one trace with a fixed time window. A spectrogram makes a plot of the frequency content of a signal trace as a function of time and is useful to distinguish different kinds
Figure 8.9: Spectral analysis

On top the original trace is seen and on the bottom the displacement spectrum (log-log, unit nm-sec and Hz). The level and slope has been indicated interactively. Note the noise spectrum at the bottom of the figure.
Figure 8.10: Three component analysis
On top the Z-channel is shown together with the window used for the 3 channels Z, N and E shown below. The signals below has been filtered between 1 and 5 Hz and the resulting azimuth of arrival is 160 degrees and a correlation coefficient of 0.2. The apparent velocity is 9.8 km/sec.
Figure 8.11: Plotting response curves
The figure shows the amplitude and phase response for station SUE, component S Z. The response is the one which will be used in analysis irrespective of whether it is taken from the file header or the CAL directory.
of seismic signals. It is often used to identify explosions, see examples below.

The built in program

In multi trace window, press ‘s’ and a spectrogram appears for each trace. The signal is by default filtered before the spectrogram is made using the predefined filter in MULPLT.DEF given by parameter SPECTROGRAM FILTER. To disable the filter, put in 0.0 in the SEISAN.DEF. In spectrogram mode, the usual commands for picking phases, zoom and selecting other traces etc. is working. To return to normal plot, use replot. In order to do spectrograms with other filters, select filter (e.g. variable filter with ‘.’) and press ‘s’. It is also possible to make spectrograms of instrument corrected signals. To e.g. make a spectrogram of acceleration, choose ‘g’ for ground motion, ‘s’ for spectrogram and ‘a’ for acceleration. This can be done whether the current plot is a spectrogram or a trace plot. When a replot of the spectrogram is done, the default filter is removed like for a replot in traceplot mode.

Figure 8.12 shows an example of a multi trace spectrogram. Note how clear the P and S-arrivals are seen on the plot. However, since the spectrograms are made doing spectral analysis in fixed time windows, the arrival times will not fit the spectrogram exactly. It would have been possible to smooth the spectrograms (or make it contoured), see the ObsPy example. However without smoothing, we see the data as it is.

NOTE: The possible output file mulplt.eps does not contain the spectrogram. To get a plot file, use the ObsPy option.

ObsPy spectrogram
8.11. PARTICLE MOTION PLOTS

The ObsPy SPECTROGRAM program in SEISAN is based on Python routines from ObsPy (docs.obspy.org) so both Python and ObsPy must be installed, see installation section. In MULPLT, the spectrogram is made in multitrace mode: Put the cursor on the trace you want to analyse at a position where the time series for the spectrogram should start. Press E (shift e) and the spectrogram will be plotted in a separate window (close the trace window if opened first). To continue with MULPLT, close the spectrogram window. The window length is by default 80 sec and a default high pass filter (2 Hz) with 2 poles is used before calculating the spectrogram. The reason for the filter is that most spectrograms are only interesting at the higher frequencies, particularly for explosion discrimination. Optionally a window with the filtered signal can also be plotted before the spectrogram comes up, see example below. The options are set up in MULPLT.DEF and the parameters used are:

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPECTROGRAM WINDOW</td>
<td>60.0</td>
</tr>
<tr>
<td>SPECTROGRAM FILTER</td>
<td>2.0</td>
</tr>
<tr>
<td>SPECTROGRAM PLOT TRACE</td>
<td>0.0</td>
</tr>
</tbody>
</table>

The window can have any value up to 300 sec (hardwired value in MULPLT). The filter is always high pass and 2 poles. If no filter is to be used, set it to 0.0. If the trace is to be plotted, the parameter must be set to 1.0.

The spectrogram program can also be operated as a free standing program. An example is:

```
spectrogram waveform.out 80 3 1
```

where the first argument is the file name, second is window length, third is filter and last is trace plotting options. All arguments must be there. In principle most waveform formats can be used (as implemented by ObsPy) but some versions of SEISAN waveform format might not work. The safest is to use MininSeed. If the waveform file contains several channels, the first one is used. Using the program independently from MULPLT, there is not the 300 s limitation.

If several tests are to be made with different window lengths and filters, the simplest way is to first make the output file waveform.out with MULPLT spectrogram option and then do the tests from the command line as the shown with the example above.

Figure 8.13 shows an example of an explosion and an earthquake at about the same location recorded at the same station. Figure 8.14 shows the corresponding spectrograms. It is clear from the seismogram which is the explosion due to the monochromatic waves (probably Rg) and the spectrograms also show more monochromatic content for the explosion than for the earthquake. Station OSL is close to the event and the explosion (epicentral distances 10-20 km). However, the same is seen for station SKAR 157-159 km away (Figure 8.15). This is a textbook example and not all explosions and earthquakes are so easily distinguished.

8.11 Particle motion plots

Particle motion plots can be made in multi trace mode when three components from one station are selected. The particle motion is plotted below the rescaled trace plot. The particle motion plots are made for the time window shown in the trace plot. The trace plot has all the normal functionality, so it is still possible to zoom and filter. The particle motion plots can be useful when determining phase types. No readings can be made from the trace plots.
CHAPTER 8. TRACE PLOTTING, PHASE PICKING AND SPECTRAL ANALYSIS, MULTLET

Figure 8.13: Top: Seismogram of an earthquake recorded at station OSL. Bottom seismogram of an explosion at the same station. Both seismograms plotted with SPECTROGRAM. The filter is 2 Hz.

Figure 8.14: Spectrograms of the two signals from OSL, explosion left, earthquake right. Note the clear difference in frequency content.
8.12 Setting MULPLT parameters, MULPLT.DEF

In the MULPLT.DEF and the SEISAN.DEF files, it is possible to set the various parameters for MULPLT. Nearly all parameters are set in the MULPLT.DEF except geometrical distance parameters, which are set in SEISAN.DEF since these parameters also are used by HYP. MULPLT will operate without DEF-files using hardwired defaults. The MULPLT.DEF can be located in the working directory and or in DAT. The if a DEF-file is present in the working directory, it overrides the file in DAT. In MULPLT.DEF, several groups of parameters can be set: The keyboard, default channels to use and analysis parameters (e.g. for spectral analysis). The parameters are identified by keywords, see example file below for explanation. Note that all numbers given in file are real and must be given a "."

Example file:

This file is for defaults for MULPLT and called MULPLT.DEF. The name must be in upper case on SUN. The following shows the parameters, which can be set. The file can contain any number of lines in any order, only the lines with recognized keywords and a non blank field under Par 1 will be read. Numbers under Par1 and Par2 must be given as reals. The comments have no importance.

This file is for defaults for MULPLT and called MULPLT.DEF. The name must be in upper case on Linux. The following shows the parameters which can be set. The file can contain any lines in any order, only the lines with recognized keywords and a non blank field under Par 1 will be read. The two parameters are in column 41-50 and 51-60, repectively. The comments have no importance. Blank means default is used. The numbers in Par 1 and Par 2 are always real numbers and anc be placed anywhere withing the 10 fields.

KEYWORD...............Comments...............Par 1......Par 2

*** Screen appearance ***

X_SCREEN_SIZE Size in pixels 90.0
NCHAN PER SCREEN # chan pr screen
Figure 8.16: Example of particle motion plot.
NSORT_DISTANCE 0: no, 1: yes
RESOLUTIONX # points pl. screen 1500.0
RESOLUTIONHC # points pl. hc 3000.0
PLOT DISTANCE 0: no 1: epi 2: hyp 1.0
#BANDPASS FILTER Default filter 4.5 9.6
CHANNEL SORTING Sort channels 1.0

*** Channel selection ***

#DEFAULT CHANNEL TRO S Z
MULPLT AREA 0.0
MULPLT LAT LON 60.0 5.0
MULPLT RADIUS 2.0
MULPLT STAT BER

*** Key assignments, see also filters for processing  ****
*** Do not use keys with fixed assignment, 1 to + are free  ****

# Example of using 1 and 2 for P and S, uncomment to use
#PHASE NAME KEY Phase key and phase 1 IP
#PHASE NAME KEY 2 IS
PHASE WEIGHT KEY Weight key and weight
PHASE MOUSE KEY Mouse key character

*** Filters for processing ***

### Set Filter type 0 for bndpas 1 for recfil routine
FILTER TYPE 0.0
FIX FILTER Fix BANDPASS FILTER, 0, 1
### Setup user-defined filters
FILTER 1 0.01 0.1
FILTER 2 0.1 1.0
FILTER 3 1.0 5.0
FILTER 4 5.0 10.0
FILTER 5 10.0 15.0
FILTER 6 15.0 25.0
FILTER 7 2.0 4.0

*** Spectral analysis ***

SPECTRAL Q0 Q0 440.0
SPECTRAL QALPHA Q = Q0**Qalpha 0.70
SPECTRAL KAPPA
SPECTRAL P-VELOCITY P velocity 6.2
SPECTRAL S-VELOCITY S velocity 3.6
SPECTRAL DENSITY Density
SPECTRAL MODEL 0 no 1 yes 1.0
SPECTRAL OUTPUT Makes a file 1.0
SPECTRAL F-BAND Spectral band to show 0.01 20.0
**CHAPTER 8. TRACE PLOTTING, PHASE PICKING AND SPECTRAL ANALYSIS, MULPLT**

*** Spectrogram ***

<table>
<thead>
<tr>
<th>Spectrogram Command</th>
<th>spectrogram.py</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spectrogram Window</td>
<td>60.0</td>
</tr>
<tr>
<td>Spectrogram Filter</td>
<td>2.0</td>
</tr>
<tr>
<td>Spectrogram Plot Trace</td>
<td>1.0</td>
</tr>
</tbody>
</table>

*** Three component analysis ***

3COMP VELOCITY velocity for 3 comp

*** Magnitude ***

<table>
<thead>
<tr>
<th>Autocoda Filter</th>
<th>0.1 10.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autocoda STA</td>
<td>3.0</td>
</tr>
<tr>
<td>Autocoda Ratio</td>
<td>1.2</td>
</tr>
<tr>
<td>CODA AUTO</td>
<td>0.0</td>
</tr>
<tr>
<td>#ML Low Cut and Poles</td>
<td>1.25 8.0</td>
</tr>
<tr>
<td>#ML High Cut and Poles</td>
<td>20.8.0</td>
</tr>
</tbody>
</table>

*** Auto processing ***

<table>
<thead>
<tr>
<th>Auto Locate</th>
<th>0,1,2 0,1,2 0.0 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Process</td>
<td>0,1,2 name 0.0 ls</td>
</tr>
</tbody>
</table>

*** Format of waveform output ***

MULPLT WAV OUT FORMAT Output format SEISAN

All parameters are within column 41 and 60 and each occupying up to 10 characters.

NOTE: If any of the phase or weight keys are redefined, all previous defaults disappear.

**DEFAULT CHANNEL**: All channels are default if not given. For routine display, it is useful to only select some channels.

**PHASE NAME KEY**: The keys associated with given phases. Remember that I, E or a blank MUST be part of the name so it is not possible to chose a name like "P", it must then be " P" (note the blank in front of P). About 10 phase combinations are currently default as seen on the pick display. If a new phase key is selected, you must define all the keys you want to use for phases including all the predefined phases. The combined onset/phase key can be up to 9 characters.

**PHASE WEIGHT KEY**: The defaults are upper case 1,2,... to 0 for weights 1,2,... to 0 . Again, choosing just one other key, and all must be redefined. The symbol must be in column 41 and the weight in column 51. The weight is an integer 0, 1, 2, 3, 4 or 9.

**PHASE MOUSE KEY**: The default is blank. Normally no redefinition is needed since the mouse character is defined in SEISAN. The key can be defined as a character or the ASCII code written as a
real number.

**SPECTRAL P-VELOCITY**: P-velocity in km/sec, default 6 km/sec

**SPECTRAL S-VELOCITY**: S-velocity in km/sec, default 3.5 km/sec Both above parameters must be set separately, the Vp/Vs in `STATION0.HYP` is not used to calculate one from the other. The values go into the S-file the first time spectra are calculated. If values are changed later in the `MULPLT.DEF` file, no change will be made in the S-file, old values remain.

**SPECTRAL Q0**: Q is defined as $q_0 \cdot f \cdot qalpha$, default 0 meaning no Q-correction. For $f \geq 1$ Hz, $Q=q0$ (new from version 10.1).

**SPECTRAL QALPHA**: See above, default 1.0, NOTE: Q is only used when doing spectral analysis and has no effect on the displacement seismograms.

**SPECTRAL DENSITY**: Density for spectral analysis (g/cm**3), default 3.5 g/cm**3

**SPECTRAL KAPPA**: Near surface attenuation, default 0.0 meaning no attenuation. For teleseismic events, this is $t^*$.

**SPECTRAL GEO DEPTHS**: Depth range where geometrical spreading changes from surface wave to body wave spreading, S-waves only. Default 50 and 100 km. This is only used if distance is larger than `HERKIJ DISTANCE`. THIS PARAMETER IS NOT SET IN `MULPLT.DEF`, BUT IN `SEISAN.DEF`, MENTIONED HERE SINCE IT IS IMPORTANT FOR SPECTRA.

**SPECTROGRAM COMMAND**: External command to compute spectrogram for single trace that can be started from MULPLT in multi trace mode (command 'E').

**HERKIJ DISTANCE**: Epicentral distance at which geometrical spreading changes from body wave spreading to surface wave spreading, S-waves only. Default 100 km. THIS PARAMETER IS NOT SET IN `MULPLT.DEF`, BUT IN `SEISAN.DEF`, MENTIONED HERE SINCE IMPORTANT FOR SPECTRA.

**3COMPVELOCITY**: Velocity used (km/sec) in 3 component azimuth analysis. Default is 5 km/sec.

**CHANNEL SORTING**: If set to 1.0, channels and filenames are sorted alphabetically, if 0.0, no sorting. Default = blank is sorting.

**NCHAN PER SCREEN**: The number of channels to be displayed per screen. Default = blank is 99 channels. It may conflict with `DEFAULT CHANNEL`.

**NSORT DISTANCE**: If blank or zero, channels are plotted in the order as they appear in the waveform file or in alphabetical order if flag `CHANNEL SORTING` is set. If set to 1.0, the channels are plotted in distance order with respect to the epicenter or if no epicenter is given with respect to the location of the station that has the first pick. If not plotted from EEV, 1.0 will indicate sorting in waveform file header time order. Default 0.

**X_SCREEN SIZE**: Size of initial X-window in % of total screen. Default 90 %.

**RESOLUTIONX** and **RESOLUTIONHC** is the number of points plotted on the screen or laser printer respectively. If e.g. 1000 points are plotted, this means that the remaining points are skipped although some primitive smoothing is done. Choosing too few points can lead to funny looking seismograms with aliasing effects and using all points will slow down the plotting. `Resolutionx` is for the screen and `resolutionhc` for the hardcopy. NOTE: If using MULPLT mode where both screen and hardcopy is used, it is the hardcopy resolution, which is used for both. Default 1000 and 3000 respectively.

**PLOT DISTANCE**: If set to 1.0 plot epicentral distance, if set to 2.0 plot hypocentral distance an dif set to 0.0, do not plot distance. Distance is only plotted on multi trace plots, just after the DC. Distance
SPECTRAL F-BAND: Spectral range (Hz) used for spectral plots. Default values are 0.05 to 20.0 Hz.

AUTO_PROCESS: Immediately following registration, MULPLT can run any program specified here. Since the event name has been put into memory, the program can operate on the newly registered S-file. Parameter one has the options: 0: Do not auto process, 2: Ask the user if autoprocess, 3: Autoprocess without asking the user. Parameter 2 gives the name of the process to run. The name is limited to 10 characters. Default, no auto processing.

AUTO LOCATE: Immediately following registration, MULPLT can locate the newly registered event and put the location into the database. Parameter one has the options: 0: Do not locate, 1: Ask the user if locate, 2: Automatically save in database. Parameter 2: 0: Do not save in database, 1: Ask if saving in database. Default, no auto locate.

SPECTRAL OUTPUT: If parameter set to 1, two output files are created for each signal spectrum. \texttt{com} \texttt{spec.out} is the complex spectrum and \texttt{amp spec.out} is the real spectrum. Default 0.0. In addition, the single trace zoom window is saved in \texttt{signal.out}.

FILTER: Change definition of filters 1 to 7. The settings affect both the shortcut keys and the menu boxes.

MULPLT WAV OUT FORMAT: The output format when using Out function, default is SEISAN. The options are the same as available in WAVETOOL.

ML LOW CUT AND POLES and ML HIGH CUT AND POLES: Filter band for Wood Anderson additional filter. Recommended values are 1.25 Hz to 20 Hz and 4 poles. Note, poles are always 4 whatever is specified.

BANDPASS FILTER: When using all defaults from EEV (option PO), a bandpass filter can be set. Default is no filter. The parameters are lowcut and highcut for parameter one and two respectively. 4 poles only.

FIX FILTER: When using all defaults from EEV (option PO), a bandpass filter can be set, see parameter BANDPASS FILTER. The filter can be fixed with parameter FIX FILTER: 1.0 = fix filter, 0 = no fix. Fixing means that in all operation the filter will remain.

CODA AUTO: Enable automatic coda determination (YES or NO). Default is NO. If enabled, auto coda is read with c instead of C.

AUTOCODA FILTER: Filter band for automatic coda: Default 5 to 10 Hz.

AUTOCODA STA: Auto coda short term average: Default 5.0 secs.

AUTOCODA RATIO: Autocoda ratio. Default 1.5.

MULPLT AREA: Options for plotting stations in a given distance from a midpoint. 0: do not select option (default) 1: midpoint from s-file epicenter, radius from MULPLT.DEF, 2: midpoint from MULPLT.DEF, radius interactive 3: midpoint and radius from MULPLT.DEF, 4: midpoint from MULPLT.DEF, radius interactive, 5: midpoint and radius interactive, 6: Midpoint from a station in MULPLT.DEF, radius from MULPLT.DEF 7: Midpoint station asked at start of MULPLT, radius from MULPLT.DEF 8: Both midpoint station and radius asked at start of MULPLT

MULPLT LAT LON: Midpoint

MULPLT STAT: Station for midpoint
8.12. *SETTING MULPLT PARAMETERS, MULPLT.DEF*

- **MULPLT RADIUS**: Radius in degrees
- **SPECTROGRAM COMMAND**: name of spectrogram program
- **SPECTROGRAM WINDOW**: window length of spectrogram
- **SPECTROGRAM FILTER**: frequency of high pass filter for spectrogram, if zero no filter is used
- **SPECTROGRAM PLOT TRACE**: if zero, no plot, if 1.0 plot trace
Chapter 9

Distance trace plot with GMT, TRACE_PLOT (Unix only)

TRACE_PLOT is a simple program to create a distance trace plot using GMT programs (Generic Mapping Tools, http://gmt.soest.hawaii.edu/). The axes of the plot are time and distance, and the traces are centered on the respective epicentral distance. The input to the program is a single event in Nordic format (S-file). From the S-file, the program reads the origin time, epicenter location and the names of the associated waveform files. TRACE_PLOT reads the waveform data and writes the x-y coordinates of the lines in the plot to a file that is then used as input to the GMT program psxy. The TRACE_PLOT program removes the DC from the data and as an option can apply a band-pass filter. The output of the program is a Postscript file (trace.plot.ps) and a batch file that can be modified and used to rerun the GMT programs (trace.plot.bat). The parameters are set in the trace.plot.par file, which can be located either in the DAT or in the working directory. An example is seen in Figure 9.1.

The parameters in trace.plot.par are:

FILTER: The pass-band filter limits can be specified through the FILTER parameter.
DISTANCE: The distance range (y-axis) for the plot.
TIME: The time range in seconds (x-axis).
AMPLITUDE_SCALE: The amplitudes are scaled for every trace individually, by [amplitude/(max amplitude) * AMPLITUDE_SCALE].
STATION_SFFILE_ONLY: This variable can be set to 1.0 to only plot traces that are listed in the S-file, the default is 0., which plots all traces without checking if they are present in the S-file.
TIME_ORIGIN: In the current version, the origin of the time axis corresponds to the origin time of the event.
COMPONENT: This can be used to select components for plotting, in case no component is defined, TRACE_PLOT will show all vertical component traces.

Example of trace.plot.par:

<table>
<thead>
<tr>
<th>KEYWORD</th>
<th>Comments</th>
<th>Par 1</th>
<th>Par 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILTER</td>
<td>filter range</td>
<td>0.1</td>
<td>5.0</td>
</tr>
<tr>
<td>DISTANCE</td>
<td>dist range</td>
<td>0.</td>
<td>440.</td>
</tr>
<tr>
<td>TIME</td>
<td>time window</td>
<td>0.</td>
<td>250.</td>
</tr>
</tbody>
</table>
AMPLITUDE_SCALE amplitude/max*scale 15.
STATION_SFILE_ONLY 0. if any station 0.
  1. if station has
    to be in s-file
TIME_ORIGIN 1.=origin time 1.
  0.=file start time
COMPONENT BH Z
COMPONENT HH Z
Figure 9.1: Example of TRACE_PLOT output.
Chapter 10

Plotting epicenters

In SEISAN there are several different programs for plotting epicenters, EPIMAP, W_EMAP and GMAP, MAPG and MAP. EPIMAP is the basic program for plotting epicenters in SEISAN and is the program that is called from EEV, when you type `map`. Program MAP is an automatic version of EPIMAP and does not require input parameters either than the hypocenter file. MAPG is a similar program to MAP using GMT. W_EMAP is a windows based program. GMAP is linked to the Google Earth and Google Map Internet mapping tool.

10.1 EPIMAP

The command for plotting epicenters is EPIMAP `file`, where the optional file is a file with EPIMAP commands. If file is not given, the user will be prompted for the input. The program can plot land contours, epicenters, macroseismic intensities, stations and level contours as well as depth profiles. It is possible to zoom in on selected areas (option by Mario Villagrán). The program has been much revised by Jim Bolton.

Input files: Land contours and other contours

The program will look for all files ending with .MAP located in the DAT directory. The user can then choose any one or a combination of files. The users own contour files (e.g. faults) can be added to the DAT directory. A very detailed world map is available on the SEISAN CD and on the SEISAN web site. Areas can be selected out of these files with program SELMAP.

Stations

Epmap will look in STATION0.HYP for station coordinates. It will search first in the working directory, then in DAT.

Epicenters

The user will be prompted for epicenter input files. The format can be Nordic or Nordic compact. Magnitudes are plotted proportional with symbol size unless the ellipticity option is selected in which case the error ellipses are plotted (if smaller than 100 km). Fault plane solutions can optionally be plotted instead of error ellipses. The first fault plane solution found in file will be used. Name of intensity files (SEISAN standard format, see Appendix A) are also entered here. The file name must have the 3 letters ‘mac’ after the ‘.’. See also section 32.
Input files for EPIMAP can be made e.g. with the COLLECT command which collects S-files into one file or with the SELECT command selecting data from the database using several criteria. HYP also generates a CAT-file (hyp.out) which may be used as input to EPIMAP.

Macroseismic information

EPIMAP can plot SEISAN macroseismic observations, see section 42.

Magnitudes

The program will read all 3 magnitudes (magnitude1, magnitude2 and magnitude3) in the header line. It will use the first non-zero magnitude in the order magnitude1, magnitude3 and magnitude2. Epimap will search the first header line only. If it is desired to use a particular magnitude from any header line for plotting, use MAG program first to select particular magnitude type which is then placed in first header line magnitude position one. Program NORHEAD can move magnitudes from following header lines to the first line. Program REPORT can move magnitudes around on the header line.

A typical run is as follows, comment after !:

Projection menu
===============
1: POLAR STEREOGRAPHIC  conformal, azimuthal
2: ORTHOGRAPHIC view from infinity, azimuthal
3: MERCATOR  cylindrical, conformal
4: LAMBERT EQUAL AREA  azimuthal
5: GNOMONIC  Great Circles are straight lines
6: AZIMUTHAL EQUIDISTANT  distance from origin is to SCALE$
7: STEREOGRAPHIC  conformal, azimuthal
8: EQUIDISTANT CYLINDRICAL cylindrical, conformal
9: OBLIQUE MERCATOR  cylindrical, conformal
10: MOLLWEIDE ELLIPTICAL  pseudocylindrical, equal area
11: SANSON’S SINUSIDIAL  pseudocylindrical, equal Area

Please enter projection number : 3
Enter latitude range of the map : 60 70 ! N positive
Enter longitude range of the map : 0 30 ! E positive

Center of geographical map space is ( 65.0, 15.0) degrees.
! can be used e.g. to make an accurate MERCATOR at high latitude
Press <return> to accept these as the reference latitude
do ordinate for the projection or <N>o to enter your own

Enter latitude of any grid line and also the grid spacing
: 60 2  ! possible to have grid spacing at any value
Enter longitude of any grid line and also the grid spacing
: 0 4

DO YOU WANT THE EVENTS NUMERATED? (Y/N=RETURN)
! a sequential number will be plotted besides each hypocenter, the

corresponding hypocenters are found in output file epimap.out.

Plot title (max 60 chars), or press <return> for none:
Plot error ellipses: e or fps: f (n=return)?: ! from HYP or fps programs
File name for contour levels, or press <return> for none: ! format below
Plot place names (P) or ! file format below
Plot all (a) or some (s) stations with a label, ! if s, question given about
! which station, give in UPPER CASE all stations
without a label (X), or none <return>
...Enter in uppercase if you wish the symbols to be filled
and this facility is available... ! only filled out in Postscript :

Available colour index values are:
1) Blue
2) Green
3) Red
4) Yellow
5) White
6) Black (default)

Enter epicenter filename and colour index,
separated by a blank, otherwise press <return>
collect.out 2 ! plot first file green

Input file is Nordic ! full Nordic format of input file

Enter epicenter filename and colour index,
separated by a blank, otherwise press <return>
select.out 3 ! plot second file red

Input file is Compact ! compact file format of input file

Enter epicenter filename and colour index,
separated by a blank, otherwise press <return>

Enter the following in uppercase if you wish the symbols
to be filled and this facility is available...
By default, symbols will be plotted according to
magnitude, do you wish them to be plotted according
to magnitude range ?

==== Loading Epicenters...

! now comes the plot, see below for options

Earthquake locations inside the window are in epimap.out
Coordinates of the surrounding area are in epimap.cor
Area selected epicenters are in epimap.are
Plot file is called epimap.eps

Interactive options:

When the plot is shown, there appears in the lower left-hand corner a menu of several options:

Q: Quit
P: Profile
A: Area
Z: Zoom

Press one of the letters to continue.

P: Profile

One or several depth section windows can now be selected with the cursor. First move the cursor to where the section shall start (from where distances are calculated), press any character to select point, move cursor to end of profile, press any character to select. A line between the two points is now plotted. Move the cursor to a point on the side of the line and press any character. A rectangle defined by the three points is now drawn, which defines the area used for the section. If more than one section is wanted (up to 9), press the number of sections instead. The selected number of profile boxes will now be plotted, all the same size. Pressing any character will draw the depth sections auto scaled, while PRESSING THE CHARACTER F, THE X AND Y SCALES ARE EQUAL and determined by the horizontal extension. When the first section appears, you can either press q to quit or any other character to plot next profile or, if the last profile, replot epicenter map and select new sections. IF YOU WANT ALL SECTIONS TO REMAIN IN PLOT FILE, QUIT AFTER PLOTTING THE LAST PROFILE. The plot file always stores what has been plotted so far, and is overwritten when a replot is made. It is also possible to plot a previously defined profile by entering O. The parameters are then taken from file profile.out. This file stores the last parameters selected by EPIMAP, but can also be edited by the user.

A: Area

Select, by clicking with the cursor, at least 3 points defining a polygon within which epicenters are selected. A new plot is made enclosing only the polygon and showing the epicenters within the polygon. The corresponding epicenters (S-files) are in file epimap.are. Known bug: Sometimes epicenters are still left outside, SELECT can be used instead.

Z: Zoom

Similar to Area, however a rectangle is selected by defining just the 2 diagonal corners.

MAP files The map files consist of blocks of coordinate pairs. Each block starts with the number of pairs in the block. The format of the header line is i4 and the following lines 10f8.3. Thus each block can at most have 9999 pairs.

Plotting place names If option P is used when the program asks for place names or station codes, the user will be prompted for one or several files with place names. The place name file format is:

name latitude,degrees longitude degrees

eg:

Edinburgh  55.94422  3.20096 or
Edinburgh 55.94422 3.20096 etc.

The only requirement is that at least 2 blanks separate the place name and the geographical co-ordinates. Note that the place name can contain one or more blanks, however each blank must occur singly. An example of a place name file is place_names.macro located in DAT.

Epimap contour file

EPIMAP has a simple contouring routine accepting a regular spaced grid. Below is an example (output from EQRSEI). The top part of the file is just comments, the data starts at "Fields to use". The data must come in longitude, latitude pairs (+ value of contour) in order as shown below. The contour value is plotted exactly as shown below. E.g. the value 117 is plotted as _117_ where "_" is blank. By specifying ___117.0, the value would be plotted as 117.0 and moved one space to the left on the plot. The example below is from an older version of SEISAN which had a program for making hazard maps. Currently the only program for making a contour file is HYP when it is used for making a gridsearch. In the DAT directory, there is an example of a gridsearch.out file made with HYP.

NB: In the input file shown below, the FIRST COLUMN MUST BE blank.

This file is input for epimap contour plotting. The input of parameters must appear as listed below with the actual data starting after the line Fields to use. There can be any number of header lines. The contour values plotted are plotted with the number of decimals given below. The last part of the file is the actual longitudes, latitudes and levels. Color use is optional and the field can be left blank, the codes are as follows:

1: blue  2: green  3: red  4: yellow  5: white  6: black

<table>
<thead>
<tr>
<th>Fields to use</th>
<th>I.......I.......I.......I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contour level to plot and color</td>
<td>117 2.0</td>
</tr>
<tr>
<td>Contour level to plot and color</td>
<td>119 3.0</td>
</tr>
<tr>
<td>Contour level to plot and color</td>
<td>121</td>
</tr>
<tr>
<td>Contour level to plot and color</td>
<td>123</td>
</tr>
<tr>
<td>Contour level to plot and color</td>
<td>125</td>
</tr>
<tr>
<td>Contour level to plot and color</td>
<td>127</td>
</tr>
<tr>
<td>Contour level to plot and color</td>
<td>129</td>
</tr>
<tr>
<td>Contour level to plot and color</td>
<td>131</td>
</tr>
<tr>
<td>Contour level to plot and color</td>
<td>133</td>
</tr>
<tr>
<td>Contour level to plot and color</td>
<td>135</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.00000</td>
<td>40.0000</td>
<td>117.620</td>
</tr>
<tr>
<td>6.00000</td>
<td>40.0000</td>
<td>118.490</td>
</tr>
<tr>
<td>7.00000</td>
<td>40.0000</td>
<td>119.080</td>
</tr>
<tr>
<td>8.00000</td>
<td>40.0000</td>
<td>119.390</td>
</tr>
<tr>
<td>9.00000</td>
<td>40.0000</td>
<td>119.390</td>
</tr>
<tr>
<td>10.00000</td>
<td>40.0000</td>
<td>119.080</td>
</tr>
<tr>
<td>11.0000</td>
<td>40.0000</td>
<td>118.490</td>
</tr>
<tr>
<td>12.0000</td>
<td>40.0000</td>
<td>117.630</td>
</tr>
<tr>
<td>13.0000</td>
<td>40.0000</td>
<td>116.510</td>
</tr>
<tr>
<td>14.0000</td>
<td>40.0000</td>
<td>115.180</td>
</tr>
</tbody>
</table>
EPIMAP output files:

epimap.out: Gives a numbered list of all events within main window. This can be used in connection with the number option.

epinmap.cor and epimap.are: If option A (selecting area) has been used, the coordinates of the corners will be given in epimap.cor and the complete events (S-files) within the selected area, in epimap.are.

epinmap.num: A compact file of epimap.out with the numbers plotted.

epinmap.eps: Postscript plot file of epicenters and possible profiles. If only one profile has been selected, all is on one page. If several profiles are selected, there will be two profiles per page up to a maximum of 6 pages (one with map and 5 with profiles).

epinmap.inp: This file is storing all input parameters of the run and can be used to run epimap again without entering any parameters. The file can be edited if a run has to be repeated with e.g. a new epicenter file. The file can have any name so several predefined plot definitions can be stored and thereby automate map production.

profile.out: The file stores the parameters used with the profiles. The file is overwritten for each new profile parameter selection. An example is:

\begin{verbatim}
17.99949 -71.54082 18.99619 -71.55782 18.44856 -71.99660
42.7
1
18.0063782 -71.0928040
17.9925880 -71.9888306
18.9893341 -72.0058365
19.0030403 -71.1098099
\end{verbatim}

The first line gives latitude and longitude of the 3 points used for selecting profile (see explanation for interactive section), next line the azimuth calculated for the profile. The third line gives the number of profiles. The following 4 lines are the coordinates (lat,lon) of the map of the profile. The file can be used to repeat the same profile as in an earlier run or to redefine a more exact profile than can be selected with the cursor. The file can also be used to plot the location of the profile with other programs.

profile.num: Output of distance and depth of the profile in km. Distance is only correct in unzoomed plots.

Problems: Known bug: When selecting events with polygon, sometimes some events remain outside.

Figure 10.1 and Figure 10.2 shows examples of plots made with EPIMAP.
Figure 10.1: An example of using EPIMAP. The top shows epicenters plotted and the bottom the first of a series of profiles. The frames on the top plot show the location of the profiles. Top left shows how many events there are in the input file and how many are plotted. Events not plotted is either because they are outside the map or because the event has no location.
Figure 10.2: An example of using EPIMAP with area selection. The top plot shows where the area is selected, while the bottom plot shows the selected area.
10.2  MAP, running EPIMAP without questions

The many options in EPIMAP are often not needed, and a bit cumbersome for routine use, so the wrapper program, MAP, will automatically calculate the input parameters and start EPIMAP. To run the program type

map file-name

where file-name can be an s-file or a compact file. If no file name is given, MAP will ask for a file name.

Optionally also fault plane solutions can be plotted, then give command

map filename f

Program MAP gets its few input parameters from SEISAN.DEF using most of the same parameters as used with the EEV option map. The parameters used are:

EPIMAP_STATIONS : plot stations from STATION0.HYP. Options are: blank: no stations plotted, a: all stations with station coded plotted, x: all stations without station code plotted. NOTE: Option s (plot some stations) cannot be used.

EPIMAP_MAP_FILE : name of map file, if blank use WORLD

EPIMAP_PROJECTION : projection number, if 0, use Mercator = 3

The program will determine the area of the map from the data file and add 0.5 degree in all directions. The latitude-longitude grid will have 5 lines for both latitude and longitude, however the smallest spacing is one degree. The commands used for the plot are given in map.inp. the file can be edited to change parameters and the plot remade with command epimap map.inp.

10.3  W_EMAP, Version Windows based map program

Program and documentation by Fernando Carrilho, fernando.carrilho@meteo.ptu
Program must be installed in addition to SEISAN.

This program was developed to be used on seismic routine processing. Its main features are the capability of allowing visualization of epicentre locations, seismic stations, error ellipses, coastlines, macroseismic data, focal mechanisms (one or many) and simplified tectonics. From the previous public version (4.1), some bugs were corrected and new features added. In particular: cartographic deformation is taken into account in error ellipses and station-epicentral path draws; some bugs on printing were corrected; compacted seisan files can be used within ‘additional events’ representation; travel time curves can now be displayed; simplified relief can be displayed (if available as a MAP file); more than one file can be used for each category layer (coastlines, tectonic, relief and places names).

The program can be integrated within the SEISAN environment, since it uses SEISAN parameter files, macroseismic files, MAP files and station/model files.

If the program is called from EEV or from the command line as W_EMAP, then it displays information contained in hyp.out file, generated by the HYPOCENT [Lienert 1994] location program, included in SEISAN, in the settable working directory.

During the first run, user is driven to edit the configuration file w_emap.def that is created in the users personal directory (SEISAN_TOP/DAT/users/{username}), where most of the program parameters can be changed.

The program can automatically detect changes in the hyp.out file so the user doesn’t need to restart the
Figure 10.3: Example for plot using w_emap (example not latest version).
10.4. MAPG, A SIMPLE AUTOMATIC GMT MAP

The program can make a standard GMT plot of epicenters, stations and fault plane solutions. The input can be standard input from a file, data base or index file. All scaling is done automatically. Both standard s-files and compact files can be input. If the program is started without argument, it will ask for input data and whether stations and/or fault plane solutions should be plotted. The program uses the standard SEIGMT program to prepare the data for GMT and the stations plotted are therefore the stations found in the input data and not all possible stations in the area. The program can also be started with input on the prompt line, but then only with input from a file like

mapg select.out

With input from the prompt line there is also more options like

mapg select.out f s

which will plot fault plane solutions and stations. All options are:

s: plot stations as triangles
S: ————————- with station code above
f: plot fault plane solutions, size scaled with magnitude
F: ————————–, fixed scale
m: plot epicenter symbol with constant size, default is proportional to to magnitude

The output is a PostScript file mapg.ps and a script file mapg.gmt. The script file can be used to do modifications to the plot that is not possible with the program. Some modifications are obvious but others require knowledge of GMT. On Linux, the file can be executed by command 'source mapg.gmt'. On windows, rename file to e.g. m.bat and execute by typing 'm'. Do not use name mapg.bat since the program would no longer run by using the mapg command, only the script if started from directory with the script.

Some parameters for the program is set in SEISAN.DEF:

PLOT_PS_COMMAND: The command used for displaying the ps file, usally GhostScript. GMT_GRIDFILE: The GMT grid file used. A reasonably detailed world file is found at

Installation

All the files are included in the distribution file w_emap.exe. To install it, you just execute this install script and make sure W_EMAP is installed under SEISMO_TOP (Usually the seismo folder). The manual will only be found in INF after installation. The current version of W_EMAP is 4.8.2 but the manual has not been updated since version 4.6. An important new option is be able to directly plot a file with many hypocenters with the command w_emap filename. Known problem: Thre must be a type 7 line in S-file in order to be able to plot a focal mechanism. Some versions of w_emap plot some of the fault plane solutions with inverted colors.
The map projection is hardwired to Mercator. The maximum horizontal size is 17 cm. The horizontal size will be reduced if the vertical size is more than 1.5 times the horizontal size. The minimum lat-lon division is 0.25 deg. The title is hardwired to Epicenters, can be changed in script.

The program uses latest version of GMT.

An example run is

c:\seismo\WOR>mapg select.out s F
1990 1213 1109 19.8 LE 60.328 5.167 0.0F BER 6 1.3 0.9CBER
1995 822 0141 6.2 L 60.389 2.590 0.0L TES 13 0.7 2.3LTE 2.7CTES 2.2LNAO
1995 1120 0401 59.1 L 60.052 5.402 1.5 TES 12 0.5 2.0LTE 2.3CTES 2.1LNAO
1996 6 6 0648 30.4 L 62.636 5.050 15.0 TES 13 0.9 2.9LTE 3.0CTES 3.0LNAO
1996 6 7 1325 29.2 L 59.846 5.130 12.0F TES 12 0.6 1.9LTE 2.2CTES 2.0LNAO
1996 625 0337 32.9 L 61.587 3.496 15.2 TES 31 0.7 2.3LTE 3.0CTES 3.2LNAO
1996 7 5 0220 46.5 L 61.287 4.829 8.0 TES 10 0.4 2.0LTE 2.2CTES 1.9LNAO
1996 713 0556 46.0 L 61.419 3.872 12.2 TES 9 0.5 1.5LTE 2.1CTES 1.6WBER
1996 718 0946 51.4 L 60.158 2.070 15.0 TES 9 1.1 1.8LTE 2.5CTES 2.1WBER
1996 726 0742 12.0 L 61.756 2.316 20.8 TES 9 0.4 1.8LTE 2.4CTES 2.0WBER
1997 812 0814 24.1 L 59.818 6.646 12.0F TES 14 0.6 2.6LTE 2.7CTES 3.4LNAO
1999 529 0031 43.6 L 62.201 4.587 15.0F TES 32 0.7 3.9LTE 3.2CTES 3.8LNAO
2000 2 1 1242 20.2 L 61.606 4.693 0.0 TES 13 0.7 2.3LTE 2.1CTES 2.3LNAO
2000 210 0617 57.4 L 61.760 4.539 6.0F TES 20 0.7 3.6LTE 2.9CTES 3.3WBER
2000 210 1740 7.2 L 61.847 4.414 2.4 TES 16 0.7 3.0LTE 2.5CTES 3.0LNAO
2002 411 1316 27.9 LE 58.202 6.203 0.0F BER 7 0.9 1.8BER 2.0CBER
2004 5 2 1924 53.9 L 59.893 6.271 5.0F TES 16 0.9 1.8LTE 1.3CTES 2.0LNAO
2004 511 0050 47.7 L 60.017 5.305 7.0F TES 16 0.7 1.8LTE 1.7CTES 2.3LNAO
2004 7 5 1809 36.9 L 57.505 7.139 15.0FDNK 5 1.0 2.2FDNK 1.9CDNK
End of s-file

Grid file to use: c:\a\gmt\newbathy.grd
Nordic input file
Return to use largest magnitude, or enter magnitude order (e.g. CLBSW)
scale: size = mag * scale (for example 0.01)
Input file is Nordic.
Number of events: 19
Number of events with fault plane solution: 6
epicenter location input file for GMT psxy/psxyz: gmtxy.out/gmtxyz.out
epicenter locations with day counter from first event gmtdays.out
station location input file for GMT psxy: gmtstxy.out
time and magnitude to be used with GMT psxy: gmtime.out
time and magnitude to be used with GMT psxy: gmtime.out
Coordinates are given as longitude - latitude
input file for psmeca (Aki and Richards convention): psmeca.out
Number of events in input file: 19
Number of events with location: 19
Latitude range to use: 57.0 63.1
Longitude range to use: 1.5 7.6
Grid size in lat and lon: 2.00 2.00
psmeca: Number of records read: 6

Number of events in input file or database: 19
Output of events in input file with location: mapg.out
Output of postscript file with map: mapg.ps
Output of gmt script: mapg.gmt

Most of the output is self-explanatory. Only events with locations are used so the number of events plotted might be different from the number of events in the input file. The events used are output in mapg.out. The plot is seen in Figure [10.4].

The script file (Windows) corresponding to the above run is:

REM Set font, size and color of title
REM
REM gmt gmtset FONT_TITLE = 14p,Helvetica,black
REM
REM Cut out desired area from grid file, area is 1.500/7.600/57.00/63.100
REM
REM gmt grdcut c:\a\gmt\newbathy.grd -R1.500/7.600/57.00/63.100 -Gmapg.grid
REM
REM Make color table, global colors
REM
REM gmt makecpt -Cglobe -T-8500/8500/50 -Z > mapg.cpt
REM
REM Plot base map, midpoint is 60.05/4.55/ horizontal size in cm is 07
REM Longitude and latitude grids are: 2.00 2.00
REM
REM gmt psbasemap -Bxa002.00g002.00 -Bya002.00g002.00 -B+t"Epicenters" -JM60.05/4.55/07c -R1.500/7.600/57.00/63.100 -K -P > mapg.ps
REM
REM Compute directional gradients from a grid
REM
REM gmt grdgradient mapg.grid -Nt1 -A225 -Gmapg.gradient
REM
REM Make the map
REM
REM gmt grdimage -JM60.05/4.55/07c -R1.500/7.600/57.00/63.100 mapg.grid -Imapg.gradient -O -K -Cmapg.cpt >> mapg.ps
REM
REM Plot coast line
REM
REM gmt pscoast -JM60.05/4.55/07c -B -R1.500/7.600/57.00/63.100 -Dh -N1/1.00p,- -K -O >> mapg.ps
REM
REM Plot epicenters, size proportional to magnitude
REM
REM gmt psxy gmtxy.out -R -J -Sc -G200/0/0 -W0.3 -O -K >> mapg.ps
REM
Figure 10.4: The GMT plot from the run above
10.5. GMAP, Plotting epicentres with Google maps or Google Earth

Google maps and Google Earth seem to quickly establish themselves as commonly used mapping tools. GMAP provides the conversion from Nordic format (s-files) to the input format required by these systems, which is the Keyhole Markup Language (KML). GMAP also convert SEISAN station and polygon files. The input format of Google Earth is described on [http://earth.google.com/kml/](http://earth.google.com/kml/). GMAP required Google Earth installed on your system to plot the output file. Download Google Earth here [http://earth.google.com/download-earth.html](http://earth.google.com/download-earth.html) (note the terms and conditions on [http://pack.google.com/intl/en/eula_print_us.html](http://pack.google.com/intl/en/eula_print_us.html)). To change the color codes see e.g. [http://html-color-codes.info/](http://html-color-codes.info/). See also chapter 3.6.

GMAP can run in three modes:

- The simple GMAP - runs in the browser with Google Map
- The advanced GMAP - runs in the desktop version of Google Earth
- The automatic GMAP - runs in the desktop version of Google Earth

The three modes are described below.

10.5.1 The simple GMAP

Type gmap in eev, a file gmap.html is created and copied to your GMAP_DIR directory. When you open the gmap.html with your browser, you will be redirected to Google Maps and a green arrow will show the epicentre. The following parameters in SEISAN.DEF are used: GMAP_DIR /home/seismo/www
GMAP_TYPE MAP [MAP, SATELLITE, HYBRID or TERRAIN]
GMAP_TYPE determines which type of map Google MAPS will use, you can choose between: MAP, SATELLITE, HYBRID and TERRAIN.
10.5.2 The advanced GMAP

This mode of GMAP is a command line version, that convert SEISAN s-files to input files for Google Earth. It also convert SEISAN STATIONx.HYP files and polygone files.

- Type `gmap` on the command line to start gmap.
- Type `gmap -help` to see the options.
- Type `gmap -stat` to convert a SEISAN station files to kml.
- Type `gmap -poly` to convert a SEISAN polygon files to kml.

In `gmap` you can:

1. Set the colour of the icons.
2. Show the error ellipse.
3. Show events as a animation over time.
4. Set the scale of the icons (default: `scale=0.2*mag**0.5`).
5. Set the type of icon used for earthquakes, explosions, probable explosions and for other events.
6. Include or exclude the S-file.
7. Rename the output file.
8. Append text in KML format to the output file (See `SEISAN.DEF` parameters below).

Example:

1. Use select to grap data from your Seisan database.
2. Run the GMAP program:

```bash
unix:/home/seismo/WOR: gmap
    INPUT FILE NAME
select.out
    Title:
    West Greenland [2000;2008]

    Number of Earthquakes : 945 Explosions : 0 Probable Explosions : 2 Other events : 2
    Output file is
gmap.kml
```
3. In Google Earth open the output file gmap.kml. See Fig. 10.5.
Figure 10.5: Example of mapping with gmap. Events in West Greenland. Note the folder and the subfolders in the Places window.
Figure 10.6: If the Earthquakes and data folder is selected in the Places window S-files can be shown.
Figure 10.7: A map can be saved as raster file (File-Save-Save Image).
CHAPTER 10. PLOTTING EPICENTERS

To make an animation for events over time use the -timespan flag. As an example explosions in the south of Norway from 1983 to 2007 can be downloaded here: [http://seis.geus.net/ber-exp.kml](http://seis.geus.net/ber-exp.kml). Press the play button at the time slider at the top of the Google Earth. Use the ruler to control how the animation is displayed (speed, days shown, etc.).

GMAP parameters added to SEISAN.DEF: Icon used for earthquake, explosion, probable explosion and other events:

- GMAP_ICON_EXPLOSION: http://maps.google.com/mapfiles/kml/shapes/star.png
- GMAP_ICON_PROB_EXPL: http://maps.google.com/mapfiles/kml/shapes/open-diamond.png
- GMAP_ICON_OTHER_EVENTS: http://maps.google.com/mapfiles/kml/shapes/square.png

Events with magnitude smaller than GMAP_ICON_MSIZE will be plotted with size of GMAP_ICON_MSIZE:

- GMAP_ICON_MSIZE: 0.5

The scale of the earthquake icons is given by GMAP_ICON_XSIZE*magnitude**GMAP_ICON_YSIZE:

- GMAP_ICON_XSIZE: 0.2
- GMAP_ICON_YSIZE: 0.5

The scale of other events is furthermore multiplied by 2. Text can be added to the KML file, see this example, note the text must be placed from character no 41 to no 120:

```xml
#GMAP_APPEND_KML <!-- lines to be appended to the gmap.kml file : -->
GMAP_APPEND_KML <ScreenOverlay id="LOGO">
GMAP_APPEND_KML <name>Info and links</name>
GMAP_APPEND_KML <description> <![CDATA[
Data is in Nordic format. The format is described in the Seisan manual at UIB.<br>
UIB: http://www.geo.uib.no/seismo<br>]]>
GMAP_APPEND_KML </description>
GMAP_APPEND_KML <Icon>
GMAP_APPEND_KML <href>http://seis.geus.net/geus.png</href>
GMAP_APPEND_KML </Icon>
GMAP_APPEND_KML <overlayXY x="0" y="1" xunits="fraction" yunits="fraction"/>
GMAP_APPEND_KML <screenXY x="-0.01" y="0.99" xunits="fraction" yunits="fraction"/>
GMAP_APPEND_KML <rotationXY x="0" y="0" xunits="fraction" yunits="fraction"/>
GMAP_APPEND_KML <size x="0.1" y="0.1" xunits="fraction" yunits="fraction"/>
GMAP_APPEND_KML </ScreenOverlay>
```

GMAP -help:

`unix:/home/seismo/WOR$ gmap -help`

The GMAP program converts Nordic format to the KML format. The output file gmap.kml can be opened with Google Earth. The program prompts for an input file in Nordic format, the input file can be compact.
The program also convert SEISAN station and polygon files.

Usage: gmap [options]
input file
Title used for kml folder

## Options ##
-help Print this list
-h Same as -help
-color Define color of epicenters [blue/green/yellow/black/white]. Default color is red
   To uses other colors see description below
-timespan Events gets timetag scroll in time domain
-nodata kml file will only contain header information
-errorellipse kml file will include error ellipse
-ellipsescolor Define color of ellipse in 8 digit hex code
-ellipseswidth Define width of the ellipse line, in pixels
-stat Station locations given in STATION?.HYP files
   is converted to KML, output is gmapstat.kml
   To change color/scale/icon edit gmapstat.kml
   and change the content of Style Id=stat
-poly SEISAN polygon files like DAT/SALVADOR.MAP
   is converted to KML, output is gmappoly.kml
   To change color/width edit gmappoly.kml and
   change the content of Style Id=poly
-out_file Define name of output file (default is gmap.kml)
-verbose Be more verbose
-version Seisan version

Scale:
The scale of the icons is set by the SEISAN.DEF parameters GMAP_ICON_MSIZE, GMAP_ICON_XSIZE and
GMAP_ICON_YSIZE, see the manual for details.

Color:
Color and opacity (alpha) values are expressed in
hexadecimal notation. The range of values for any
one color is 0 to 255 (00 to ff). For alpha, 00 is
fully transparent and ff is fully opaque. The order
of expression is aabbggrr, where aa=alpha (00 to ff);
bb=blue (00 to ff); gg=green (00 to ff);
rr=red (00 to ff). For example, if you want to apply
a blue color with 50 percent opacity to an overlay,
you would specify the following:
<color>7fff0000</color>, where alpha=0x7f, blue=0xff,
green=0x00, and red=0x00. See also:
http://code.google.com/apis/kml/
documentation/kml_tags_21.html#color

Examples:
gmap -color blue -nodata -errorellipse
  gmap -timespan -color 7ee000ee
  echo "collect.out\nDK events\n" | gmap -out_file dk.kml
CHAPTER 10. PLOTTING EPICENTERS

10.5.3 The automatic GMAP

There are two way to make maps with Google Earth or Google Maps from EEV automatic.

One way is to call GHYP in EEV, this will start Google Earth or Google Maps (in a browser) automatically for each of the following events. Note, that Google Earth will reload every new event but Google Maps will open a new browser tab for every new event (tested in Chrome).

The other way is to enable the GMAP_AUTO parameter in SEISAN.DEF, this will enable a function in the HYP program, that makes a KML file that Google Earth can load automatically when e.g. an event is located in EEV. See description below.

Start EEV and locate an event. In the directory a new file gmap.cur.kml will appear, this file can be opened with Google Earth and it will show the location of the current event and the used stations and the raypaths.

The color of the stations is given by the travel time residual as green, yellow or red, for: residuals < 0.5s, 0.5s = < residual = < 1.5s or residual > 1.5s, respectively.

To enable the automatic generation of Google Earth input files, add these parameters to your SEISAN.DEF file:

GMAP_AUTO 0: no, 1:yes 1.0
GMAP_AUTO_ICON_COLOR ff0000ff
GMAP_AUTO_ICON_MSIZE 0.5
GMAP_AUTO_ICON_XSIZE 0.2
GMAP_AUTO_ICON_YSIZE 0.5
GMAP_AUTO_LOOKAT_ALTITUDE 2000000.0
GMAP_AUTO_SHOW_STAT 0: no, 1:yes 1.
GMAP_AUTO_ERROR_ELLIPSE 0: no, 1:yes 1.
GMAP_AUTO_ELLIPSE_COLOR ff00ff00
GMAP_AUTO_ELLIPSE_WIDTH 12.43
GMAP_AUTO_STAT_SIZE 1.1
GMAP_AUTO_URL http://maps.google.com/mapfiles/kml/shapes/triangle.png
GMAP_AUTO_STAT_RESIDUAL_GOOD 0.5
GMAP_AUTO_STAT_RESIDUAL_BAD 1.5
GMAP_AUTO_STAT_COLOR_GOOD ff00ff00
GMAP_AUTO_STAT_COLOR_OK ff00ffff
GMAP_AUTO_STAT_COLOR_BAD ff0000ff
GMAP_AUTO_SHOW_OLD_LOCATION 0: no, 1:yes 1.
GMAP_AUTO_OLD_LOCATION_COLOR ffff0000
GMAP_AUTO_SHOW_PATH 0: no, 1:yes 1.
GMAP_AUTO_PATH_COLOR ff929292
GMAP_AUTO_PATH_WIDTH 2.5
GMAP_AUTO_FILE_ACTION 0: no, 1:yes 0.
GMAP_AUTO_ACTION cp gmap.cur.kml /inetpub/www/html/seismo/nnsn

You can change the parameters to adjust the output file gmap.cur.kml for your system, see definition here 3.12.

In order to run the display in a realtime mode, you must have an other KML file that makes Google Earth reload the gmap.cur.kml file at short intervals. Such file could look like:

<?xml version='1.0' encoding='UTF-8'?>
Figure 10.8: Example of automatic mapping with gmap. Earthquake in West Greenland. Blue dot is the old location and the red dot is the current location. Green and yellow stations are stations with good and ok residuals, respectively.
In this example the gmap.cur.kml file is reloaded every 3 second from a local C:\seismo\WOR directory on a windows pc. And below is an other KML file is reloaded at 60min intervals from the Internet. The above file is named gmap-automatic.kml and is found in the DAT directory.

10.6 SEIS2VIEWER, Plotting hypocentres in 3D

The program is written by Ruben Soares Luís (ruben.so.luis@gmail.com) and use the SeismicityViewer50 program by Anthony Lomax. To program has been aliased to smap.

Overview

seis2viewer is a wrapper for the application SeismicityViewer, developed by Anthony Lomax for rapid mapping of seismic events. SeismicityViewer displays hypocenter locations in 3D as well as station locations and P/S residuals. Geographic and geologic features can also be displayed as well as focal mechanisms. Please find details here: http://alomax.free.fr/seismicity/

seis2viewer has been created to take information from a nordic file from Seisan and convert it in a format usable by SeismicityViewer 5.0 (NLLoc Format). It automatically launches Seismicity Viewer, generating a set of files required for its operation.

In its current version, seis2viewer allows the visualization of hypocenters and magnitudes in SeismicityViewer 5.0. Other information, such as station locations, P and S residuals, etc. is not displayed.

Configuration
10.6. **SEIS2VIEWER, PLOTTING HYPOCENTRES IN 3D**

1: **Configuration Files:**

- **seis2viewer.def:** seis2viewer uses a single configuration file: *seis2viewer.def*, which may contain references to other files containing information on maps, elevation, localities, etc. The file *seis2viewer.def* is actually passed directly to Seismicity Viewer adding only a reference to an automatically created file. This operation is transparent to the user. As such, the configurations can be obtained from the Seismicity Viewer 5.0 website: [http://alomax.free.fr/seismicity/](http://alomax.free.fr/seismicity/). The *seis2viewer.def* can be in local directory or DAT. The program will operate without the def file.

- **Other files:** *seis2viewer.def* may reference other files containing map information, elevation data, locality names, etc. These files should follow the definitions presented in the Seismicity Viewer 5.0 website.

2: **Installation**

seis2viewer is distributed as a single .jar file (*seis2viewer.jar*), containing all the necessary classes to work, including Seismicity Viewer 5.0. This file is placed in the PRO directory of seisan. To facilitate the usage of seis2viewer, an executable script file to call seis2viewer has been created, *smap*. As an example, the executable script should contain the following line:

```
java -jar %SEISAN_TOP%/PRO/seis2viewer.jar $1  (for windows)
java -jar $SEISAN_TOP/PRO/seis2viewer.jar $1   (for unix/linux)
```

The configuration file, *seis2viewer.def*, and any other required files should be placed in the DAT directory of seisan. Alternatively, the user may have its configurations on the working directory. The configuration file refers to a more detailed map file, *europe.xyz*, which will be plotted superimposed on the built in word map (can be turned on and off on the plot). The location of this file must be given an absolute path. In the example file, the path is given for Windows as

```
seismicity.lines.gray = C:/\seismo/\DAT/\europe.xyz
```

where the '/' is needed under Windows. The user must adjust this line to the local environment. Detailed map files in xyz format, using standard SEISAN MAP files, can be made with program SELMAP.

2: **Automatically Generated Files** seis2viewer generates a set of files to be used by Seismicity Viewer 5.0. Although these files are, in principle, irrelevant to the user, it is nevertheless important to mention them for reference purposes.

- **input-file.hyp:** This file contains a translation of the event file selected by the user for visualization in NonLinLoc format, appropriate for Seismicity Viewer. Input-file is the input file name without extension.

- **input-file.hdr:** This is an automatically generated grid file for the area surrounding the events to be visualized. It is only created if the maximum distance in longitude or latitude between events is less than 8 degrees.

**Usage**

seis2viewer can be used directly with a nordic file containing one or more events (e.g. *select.out*) as:

```
java -jar %SEISAN_TOP%/PRO/seis2viewer.jar select.out  (windows)
java -jar $SEISAN_TOP/PRO/seis2viewer.jar select.out   (unix/linux)
```
or using the command 'smap')

```
smap select.out
```

seis2viewer can also be called directly from eev, as an external program, using the prefix 'o' as:

```
osmap
```

In this case, smap will try to find the file eev.cur.out, which is automatically generated by eev and contains a reference to the event that is currently under work.

Seis2viewer will use a global mode if the distance between events is more than 8 degrees. The global mode can be forced with the flag -gl (but not with command smap, use complete command like

```
java -jar C:\Seismo\PRO\seis2viewer.jar -gl select.out
```

Seis2viewer normally requires a Nordic file for input and will only plot the header line hypocenter. To plot all hypocenter lines in the s-file, use option -ia. This option can also be used to plot a compact file (only header lines, no space between lines).
Chapter 11

Searching in the database, **SELECT** and others

There are several programs to search the database or search in files. The most important is **SELECT** when working from the prompt. The most sophisticated search can be made with SeisanExplorer (see SE section). **SE** and **SELECT** have overlapping functions and both have options not found in the other.

### 11.1 SELECT

Whenever selective search and extraction is wanted **SELECT** is used. The program can run on the CAT database, single CAT files (Nordic or Nordic compact) or the S-file database. The output file, `select.out`, will also be in Nordic format. Since the input CAT database can contain both normal and compact files, the output will always be a normal file with blank lines between events. If however the input is one compact file, the output will also be a compact file. Note: If **SELECT** is used on the CAT database (normal operation), you need to UPDATE your S-file database in order to transfer changes from the S-files to the CAT database. Select can work with input in 3 different ways:

1. The user is asked for selections
2. The selection parameters are in a file
3. Parameters are given on the prompt line

The program is started by typing **SELECT** (parameters from screen), **SELECT 'input file'** (parameters from input file) or **SELECT -options**. A typical user interactive run is shown below. Comments following ![POSSIBLE INPUT IS:](image)

```
STANDARD CAT DATABASE: RETURN
ALTERNATIVE DATABASE, GIVE 1-5 LETTER CODE:
FILENAME FOR ONE FILE, MUST BE 6 OR MORE CHARACTERS:
Type of base: CAT (Return) or Sfiles (s):
```
Updating database TEST_! standard base
The database TEST_ has 10 files
The first file starts: 199309
The last file starts: 199909

Start time (blank is 1980), yyyymmddhhmmss: 199309! time range
End time, enter for 2020 : 199607

PARAMETERS

1  - Fault Plane Solution
2  - Earthquake Felt
3  - Magnitude Type(s)
4  - Distance ID(s)
5  - Event ID(s)
6  - Magnitude Limits
7  - Latitude Limits
8  - Longitude Limits
9  - Depth Limits
10 - RMS Limits
11 - Number of Stations Limits
12 - Hypocenter Errors Latitude Limits
13 - Hypocenter Errors Longitude Limits
14 - Hypocenter Errors Depth Limits
15 - Minimum Number of Polarities
16 - Hypocenter Agencies
17 - Magnitude Agencies
18 - Station Codes, components and distance range
19 - Polygon
20 - Use all header lines
21 - Search for text string in S-file
22 - Gap range
23 - Phases
24 - Volcanic subclasses

SELECT NUMBER TO CHANGE PARAMETER, RETURN TO SEARCH: 6
Minimum Magnitude, return for default:  5
Maximum Magnitude, return for default:  7

PARAMETERS

1  - Fault Plane Solution
2  - Earthquake Felt
3  - Magnitude Type(s)
4  - Distance ID(s)
5  - Event ID(s)
6  - Magnitude Limits 5.0 7.0
7  - Latitude Limits
8 - Longitude Limits
9 - Depth Limits
10 - RMS Limits
11 - Number of Stations Limits
12 - Hypocenter Errors Latitude Limits
13 - Hypocenter Errors Longitude Limits
14 - Hypocenter Errors Depth Limits
15 - Minimum Number of Polarities
16 - Hypocenter Agencies
17 - Magnitude Agencies
18 - Station Codes, components and distance range
19 - Polygon
20 - Use all header lines
21 - Look for wave form file names
22 - Gap range
23 - Phases
24 - Volcanic subclasses

Ok. Input realized successfully.

SELECT NUMBER TO CHANGE PARAMETER, RETURN TO SEARCH:

SELECTED EARTHQUAKES ARE IN FILE: select.out
LOCAL INDEX FILE IN: index.out
NAMES FOR WAVEFORM FILES IN FILE: waveform_names.out
SELECT COMMANDS IN FILE: select.inp

Note above, that the second time the menu is shown, the choice of magnitude limits is shown. For each CAT file in the catalog, the number of events in file, number of events selected from that file and the accumulated number are listed. The last file might not show the correct number of events in file since SELECT might stop before reading the whole file if the end time is in the middle of the file. If start time is blank, 1980 is used. The end time can also be blank, and 2025 is used. This option is useful when selection on whole data base or whole file.
NOTE on time interval: If selecting from S-files or the CAT data base, selection will stop when one event reach the time limit so a wrong time in S-file could stop search. When selection from a file, all events in file will be checked so there is no requirement for correct time order.

Index.out file: It contains the indexes of files selected and can be used to access theses file only like cee index.out. NOTE that the index file will only contain events which have an ID line since ID line is used to create the reference.

11.1.1 Input parameters for SELECT

In the input database (or file) a time window must always be given. If no more selection is done, all data in time window is selected. Further selection can be done by choosing a number and giving parameters. The chosen parameters are then shown on the next parameter selection menu as shown above for magnitude. Parameters can be reentered. Parameters not entered will have no influence in the selection. If several parameters (numbered selections below) are entered, conditions for all must be true for the event to be selected. Within each numbered selection, usually only one of the entered conditions must be fulfilled for the event to be selected. If e.g. Ml and Mb are selected, events, which have either magnitude, will be selected. When no more parameters are desired, press enter.

1. - Fault Plane Solution
   Selects events with a fault plane solution (F- line in S-file). There will also be the question: "Give quality, e.g. A or ABC, enter for all", in this way different qualities can be selected.

2. - Earthquake Felt
   Events felt indicated by a type 2 line

3. - Magnitude Type(s)
   Normally, all magnitudes for one event are searched to see if any magnitude fits the selection criteria. With option 3 it is possible to use one or a combination of magnitude types e.g. L and B. If magnitudes without type are to be selected, use underscore “_” for magnitude type. If there is no magnitude in the first magnitude position, chose “N” for one of the magnitude types to be able to select the other 2 magnitudes on the line. Magnitude types are: C: Coda magnitude, L: Local magnitude, b: mb, B: mB, s: Ms, S: MB and W: Moment magnitude. N: Find events with no magnitude in first position. An event is selected if any one of the types of magnitudes are found. Magnitudes are only searched on first header line unless “Use all header lines is set”.

4. - Distance ID(s)
   Restricting the search to be for one or a combination of the distance id’s L, R and D.

5. - Event ID(s)
   Restricting the search to one or a combination of event id’s, e.g. E and V for explosion and volcanic events. The letters used for selection are not limited to the examples shown above, they are however the ones used currently. It is thus e.g. possibly to label events as X for unknown type (column 23 in header line) and then later on select out all those events by specifying X for event ID. For the 3 questions about types, up to 5 letters can be used. The currently most used codes are: B: Blank (no ID), E: Explosion, P: Probable explosion, V: Volcanic, S: Sonic boom, Q: Earthquakes. For a full list, see the Nordic format in Appendix. If no selection is given, all events are selected. If e.g. Q is used as input, only events with Q ID are selected. If B is selected, only events with no ID are selected. So if all earthquakes and volcanic events are to be selected, use QV. Without the Q, only volcanic events are selected. In most cases no ID is also earthquakes, but they have not been
marded as so. In this case, in order to get all earthquakes, use BQ. Selection is made if either one of criteria is met.

Note: Before October, 2018, using Q was equivalent to using BQ and there was no possibility to select Q or no ID only.

6. - Magnitude Limits
   Range of magnitudes to select. Note that if no magnitude type is given, the extreme of all magnitude types reported is used. Magnitudes are only searched on first header line unless Use all header lines is set.

7. - Latitude Limits
   Range of latitude. NOTE: If no latitude or longitude values are chosen, SELECT will include an event even when it is not located if the remaining criteria are OK. If it is required that only located events are searched for, enter at least one value like an upper latitude limit of 95.

8. - Longitude Limits
   Range of longitude.

9. - Depth Limits
   Range of depths.

10. - RMS Limits
    Range of rms travel time residuals.

11. - Number of Stations Limits
    Range of number of stations. This option can be combined with option CODAQ to search for a number of stations in a given distance range.

12. - Hypocenter Errors Latitude Limits
    Range of hypocenter latitude errors. Works only if error line (E-type) is present in S-file. Currently error lines are generated by HYP and the ISC conversion program ISCNOR. There should only be one error line in file associated with the prime solution in first header line. However, if more than one error line is present, all are checked and if one fulfills the selection criteria, the event can be selected.


15. - Minimum Number of Polarities, only P-phases are used
    Counts all polarities, useful to find potential events for fault plane solutions.

16. - Hypocenter Agencies
    Selects events only with given hypocenter agencies as indicated on header line.

17. - Magnitude Agencies
    Select only events with given magnitude agencies as indicated on header line. Magnitudes are only searched on first header line unless Use all header lines is set.

18. - Station Codes, components, distance range and phase
    Selects only events with given stations, component, distance range and phase. A formatted help line comes up for selecting items. Any one or a combination can be selected, however, a station code or component code must be selected. The phase can be 8 characters long. The distance can be hypocentral or epicentral. Distances are integers right justified.
19. - Polygon
   Selects events within a given polygon of at least 3 latitude-longitude pairs.

20. - Use all header lines
    All header lines are searched for relevant information

21. Search for text string in S-file
    Search the database for any text string in any line of the S-file. The search is normally case
    sensitive, but if the search string is terminated with ‘case’ like ‘bergencase’, then the search is case
    independent. The search string can be 80 chars long plus 4 if terminated with ‘case’. It is thus
    possible to make a search including the line type in character 80.

22. - Gap range
    The range of gap as given on the E-line (normally 2. header line). Only hypocenters calculated
    with SEISAN version 7.0 have gap.

23. - Phase
    Look for events with particular phases. Up to 6, 8 character phase names can be selected. The
    event is selected if at least one of the phases is present for the event. For a more selective selection
    based on phase, see option 18.

24. - Volcanic subclasses
    Search for events of given subclasses given by up to 10 codes. Any code can be given, however,
    normally they will be as defined in VOLCANO.DEF. The program searches for lines starting with
    ‘VOLC MAIN’.

25. - Moment tensor solution
    Search for events with a moment tensor solution as indicated by M-lines.

11.1.2 Option for codaq

A special option is to make a file used for input to CODAQ using option 18. The station name CODAQ
is selected and all stations present within the specified distance range are selected and the output is written
in file index.codaq which then contains the event-station combinations used as input to CODAQ. For
component selection, only the orientation code can be used. In addition to to Z, N or E, also A can be used.
This will select all 3 components for a selected station. In example below, all components for stations
with 100 km are selected.

STAT CO Mindis Maxdis Phas All stat hdist->FF
CODAQ A 0 100

The CODAQ option can also be used to select events in given distance range with a given number of
stations if the number of station option is also used.

Spectral parameter selection

There is no direct option to select events with spectral parameters except to select local events with
the MW and assumed they have been calculated with spectra. A possibility is to use MAG where events can
be output having particular spectral parameters. Another option is to use SeisamExplorer. /IndexSearch
for spectral parameters
11.1. SELECT

11.1.3 Historical data

When working with historical data, it can be useful to work with catalogs of several centuries. The century is available in the Nordic Format, so catalogs can go back to year 0. Output:

**select.out**: A CAT-file or compact file (depending on input) of selected events.

**index.out**: A list of event id’s of selected events can be used with EEV or other programs accepting index files. This could be used e.g. to work on only distant events in the database by first selecting all distant events and then working with these directly in the database using command EEV index.out. Index files can have any name (must contain a ‘.’) so different subsets can be available with different index files.

**Waveform_names.out**: A list of corresponding waveform files. It is mainly intended for copying to or from tape specific waveform files. It has the format of the `filenr.lis` files and can be used directly with e.g. MULPLT. See also program get wav for selecting waveform files from the database.

**select.inp**: A file with all the parameters used for the run. The file can be renamed, edited and used as input for select. This is particularly an advantage if a complex set of selection parameters are used and the selection is wanted again with just a small change. An example file is shown below

```
Base or file name : TEST_
Start time : 19930300000000
End time : 19961231235959
Minimum number of stations : 0
Maximum number of stations : 999
Minimum latitude : -90.000
Maximum latitude : 90.000
Minimum longitude : -360.000
Maximum longitude : 360.000
Minimum magnitude : 5.000
Maximum magnitude : 7.000
Magnitude agencies :
Hypocenter agencies :
Minimum rms : 0.000
Maximum rms : 999.000
Minimum depth : -99.000
Maximum depth : 99999.000
Minimum error in latitude : 0.000
Maximum error in latitude : 99999.000
Minimum error in longitude : 0.000
Maximum error in longitude : 99999.000
Minimum error in depth : -99.000
Maximum error in depth : 99999.000
Magnitude types (L,C,B,S,W) :
Distance (ID) types (L,R,D) :
Event types (e.g. E,V,P) :
Minimum number of polarities : 0
Felt earthquakes : F
Fault plane solution : F
Check all header lines : F
Waveform files to check :
```
Minimum gap : 0.000
Maximum gap : 360.000
Phases : P SSS PP
Volcanic subclasses :
Stat., comp. dist range, phase (1x,a5,a2,2i7,1x,a4) one pr line, end blank line:
STAT CO Mindis Maxdis Phas All stat hdist->TT
BER SZ 1 999 P

Polygon points (lat,lon), one pair pr line, end with blank line :

Note: The TT at STAT line indicates that all stations must be present (True) and hypocentral distance is used (True)

11.1.4 Select with input from the prompt line

This option is particular useful when using select with automated operations and has been made specifically to deal with extracting data out of the data bases using WEB based software. This option do not have all of the above options. The following are implemented:

- base : 5 letter data base
- seisweb: if set, WEB output parameters
- time : time interval (2 variables)
- web_out: complete path to where data is placed, only active if seisweb set. 3 files are made:
  - web_out.id: id's, like index.out without
  - web_out.all: like select.out
  - web_out.head: header lines
- area : lat-lon grid, minlat,maxlat,minlon,maxlon
- depth : depth range, mindepth,maxdepth
- mag : magnitude range, minmag,maxmag
- nstat : range of number of stations, min,max
- gap : range of gap, min,max
- rms : range of rms, min,max
- magtypes : up to 5 mag types, one string, e.g L
- disttype : distance type, e.g D
- eventtype : event type, e.g E

Problems: An event might be found and listed in index.out, but when looking for it with EEV, it is not there. This can happen if an event has been deleted with EEV and no UPDATE has been made, so that the event is still present in the CAT part of the database.

11.2 Searching for text string in nordic files, SELECTC

The command SELECTC is used to search for text stings in nordic files like collect.out or select.out. Events with the matching text string is listed in the output file selectc.out. The program is written by Ruben Soares Luis. Below is an example :
selectc
Input file: collect.out
Search comment: Bergen
Found 634 events. Saving output file: selectc.out

11.3 Getstressdrop

Written by Bladimir Moreno, August, 2014, bladimir@cenais.cu

This is a program to search in a SEISAN Nordic file (S-file) for the calculated stress drop, make average stress drop and Mw for each event using all channels or selected channels as well as average for the whole data set of several events. Only channels with a stress drop of 1 bar or more is used and the upper limit of stress drop to use in average is user selectable. The output is the average stressdrop for each event versus magnitude Mw, average stressdrop for the whole data set and a statistics of the stress drop distribution. In addition events can be selected in a lat-lon polygon.

In SEISAN stress drops are calculated with either MULPLT in a manual or semiautomatic way, automatically with AUTOSIG and automatically with AUTOMAG (a simplified AUTOSIG) either directly or through EEV command av. In each S-file, the average stress drop (as well as the average of other source parameters) is calculated when the event is updated, irrespective of the value of the stressdrop. The stressdrop is a good indicator of the quality of the spectral fitting and stress drops less 1 and higher than 200 bars usually indicate an unreliable spectrum. The program GETSTRESSDROP will therefore be useful for making reliable average stress drops using only the best data and channels.

The user calls the program with command line arguments. The first argument is obligatory and the rest are optional. The program is called in the following way:

gestressdrop <nordic.inp> [-c <stress_cut>] [-s <stationslist_file.inp>] [-p <points_polygon.inp>]

Nordic.inp: an S-file, any name.

stress_cut: the maximum stress drop to consider in the searching process.

stations.lis: a file (any name) with the station-channels list, with the same format as written in the line SPEC of the S-FILES format, see example later.

points_polygon.inp: a file (any name) with longitude, latitude pairs. The last point of the polygon must be equal to the first (close polygon), see example below.

20.0 -76.7
20.0 -75.0
19.5 -75
19.5 -76.7
20.0 -76.7

Output getstressdrop_histo.out: number of stressdrops in steps of 10 bars, can be plotted with LSQ.

gestressdrop_mw.out: list of earthquake magnitudes Mw versus stressdrop, can be plotted with LSQ.

Example run

gestressdrop eev.out
CHAPTER 11. SEARCHING IN THE DATABASE, SELECT AND OTHERS

1996 6 6 0648 23.8 L 63.005 3.955 8.0 TES 5 2.9 2.8LTES 2.7WTES 3.0LNAO1
SPEC FOO S Z MO 13.5 ST 26.2 OM 1.5 f0 12.7 R0.1754 AL 0.00 WI 10.0 MW 2.9 3
SPEC HASK S Z MO 13.0 ST113.0 OM 1.7 f0 17.5 R0.0738 AL 0.00 WI 20.0 MW 2.6 3
SPEC EGD S Z MO 12.9 ST 24.1 OM 0.6 f0 20.0 R0.1114 AL 0.00 WI 10.0 MW 2.5 3
SPEC AK S Z MO 13.0 ST113.0 OM 1.7 f0 17.5 R0.0738 AL 0.00 WI 20.0 MW 2.6 3
SPEC EGD S Z MO 13.1 ST 35.7 OM 1.8 f0 10.9 R0.1185 AL 0.00 WI 20.0 MW 2.7 3
SPEC BLS5S Z MO 13.2 ST 5.8 OM 1.8 f0 5.66 R0.2281 AL 0.00 WI 20.0 MW 2.7 3
1996 6 7 1325 29.2 L 59.846 5.130 12.0F TES 12 0.6 1.9LTES 2.2CTES 2.0LNAO1
SPEC EGD S Z MO 12.0 ST 12.8 OM 1.1 f0 20.0 R0.0690 AL 0.00 WI 20.0 MW 1.9 3
SPEC KMY S Z MO 12.2 ST 23.5 OM 1.2 f0 20.0 R0.0690 AL 0.00 WI 20.0 MW 2.1 3
SPEC HASK S Z MO 12.2 ST 23.5 OM 1.2 f0 20.0 R0.0690 AL 0.00 WI 20.0 MW 2.1 3
SPEC ODD1S Z MO 12.3 ST 27.8 OM 1.2 f0 20.0 R0.0690 AL 0.00 WI 20.0 MW 2.2 3
SPEC BLS5S Z MO 12.2 ST 23.1 OM 1.1 f0 20.0 R0.0690 AL 0.00 WI 20.0 MW 2.1 3
SPEC SUE S Z MO 12.4 ST 32.8 OM 1.1 f0 20.0 R0.0690 AL 0.00 WI 20.0 MW 2.2 3
SPEC HYA S Z MO 12.3 ST 29.3 OM 1.0 f0 20.0 R0.0690 AL 0.00 WI 20.0 MW 2.2 3
SPEC FOO S Z MO 12.5 ST 23.9 OM 1.1 f0 16.4 R0.0842 AL 0.00 WI 20.0 MW 2.3 3

Average stress drop for 2 earthquakes selected: 30.0

Total number of channels used: 14

File with stress drop histogram is getstressdrop_hist.out
File with stress drop as a function of magnitude mw is getstressdrop_mw.out

Each selected channel is printed out and in the end there is information of the total number of events and channels used.

An example of a channel selection file stat is:

HASK1S Z
HYA S Z

It could also be written

HASK1S Z
HYA S Z

Note that station names with 5 characters on SPEC line uses the 5. character on the line for the first character in station name.

Stress drop versus magnitude can give an indication if stress drops are reliable. According to the theory of self similarity (see e.g. Havskov and Ottemøller [2010]), stress drops should be magnitude independent. For small earthquakes (M < 3) it might be difficult to get a reliable corner frequency, it will often be to small due to near surface attenuation (kappa, see MULPLT section). This results in smaller stress drops for smaller events than larger events. It is therefore useful to plot the stress drop versus magnitude. Figure 11.1 shows an example made with program LSQ.

ndexHistogram, stress drop

It is seen that there is only a slight tendency to increasing stress drop with magnitude indicating reliable stress drops.

The distribution of stress drops can also be plotted with LSQ. Figure 11.2 shows an example of the data from 11.1.
Figure 11.1: Stress drop versus Mw for 23 events.
Figure 11.2: Histogram of stress drops.
Chapter 12

Extracting events from the database, COLLECT

The command COLLECT is used for collecting many event files from the database S-files into a single file. This may be split into individual event files later using SPLIT. The file can be used for exchanging data with other agencies or be used with the epicenter plotting program. The questions are:

Base name, ,, for local directory, name of index file
or return for default base:

Start time :
End time, return for end of month:
Compact output file (Y/N=default)

If a local data base is input, default start time is 1980 and default end time 2015. In this way it is fast to collect all data from a local data base. At the end, the program will give statistics of collected data, and file name. For getting data out of the database represented by the monthly CAT files, use SELECT. If an update has been made, SELECT will always be the fastest program to use. COLLECT and SELECT are the only programs that can make a CAT file from the individual S-files. Program input can also be on the prompt line, below is an example:

```
collect -start_time 19910912 -end_time 19911015 -base_name BER -compact
```

This means that a CAT-file (default) is collected from BER and is written in compact format (-compact has no arguments). The time interval is between 19910912 and 19911015. Only start_time is required, the other arguments are optional. The syntax is: -"keyword" value -"keyword" value etc.
Chapter 13

Inserting events into the database, SPLIT

The program splits up a multiple event S-file in Nordic format (usually made by COLLECT or NEWEVE) or compact file to single files in the database or in the users own directory. Type SPLIT to start program and questions are:

INPUT FILE NAME ! Give file name
COLLECT.OUT ! File name
INPUT BASE NAME: ! Indicate which type
BER FOR STANDARD DATABASE: ! - database
THREE LETTER CODE FOR OTHER BASE
RETURN FOR SPLIT UP IN LOCAL DIRECTORY
BER ! Choice was standard base
OPERATOR ID, MAX 4 CHARs ! Operator id logged in file
jh

1988 2 5 13:51 35.0 L RECORDS: 4 ! Listing of events split up
File already exists, options are: ! try to make a file with same
Ignore (leave old event): Return ! id
Ignore all I
Overwrite duplicate: O
Overwrite all duplicates A
Create a new event, different ID: N
Create new events for ALL duplicates: * O

1988 2 5 14:15 25.2 D RECORDS: 4
1988 2 5 19: 4 10.0 D RECORDS: 3
1992 11 1 1:32 1.0 D RECORDS: 55
In the above example, there was already an event in the database with the same file name and therefore the same id. It is up to the user to decide if this is the same event in which case it should be ignored or if it is a new event which happens to have the same id (start time or origin time to the same second and same event type). In case of a new event, a new id with one second different will be tried. Sometimes it can be desirable to overwrite the whole database event by event. If e.g. a station code is wrong in all events, this can be corrected by making a collect to extract all events, edit the collect.out file using a global substitute, and finally use split to put the events back in. In that case the option of overwriting all should be chosen.

Compact files can also be split up. Since this is unusual to do, the user will be prompted 2 times to confirm the split up. Since there is no ID line in a compact file, the database name will be generated from the header time. This option to be able to split up compact files has been made to facilitate work with seismic catalogs where it is often desirable to be able to access individual events even when no readings are available.
Chapter 14

Updating final locations in database, UPDATE and UPD

UPDATE

Both the monthly epicenter files in \SEISMO\REA\BER\CAT and the updated S-files are generated with program UPDATE which is a special version of HYP. Type UPDATE to start the program and there will be questions about time period and database. The program will also ask for operator ID (4 chars), which is stored in the updated log file and the S-file, see below.

By updating, both the S-files and the CAT-files in the CAT-directory are updated. The reason for updating both at the same time is to ensure that there is a correspondence between the two.

The program will go through as many months as specified by the user. When the program is running, one line will be printed out for each event. The S-files will be overwritten with the updated location, residuals etc. At the same time, a monthly CAT file is created in the CAT directory containing all events, also events not located. If a monthly file is already present, it is overwritten.

Update can also work on a local database. The S-files are updated as described above. Since there is no CAT database, the Update program makes a CAT file in the local directory called hyp.cat with events in chronological order.

The ID line is updated the action UPD, the operator ID and the time. At the same time, all the error lines are deleted and only the one belonging to the prime location is kept.

The update process can also change all S-file names according to the origin time and the ID’s are changed correspondingly. This is done in order for the database to be in chronological order according to origin time and not the more random times used when the events were first registered into the database. E.g. a teleseismic event is put in with the ID corresponding to the recording time, when located, the origin time is many minutes before and it will appear too late in the database). Even if the event is marked not to be located with a * in header line column 45, the ID will still be updated (same for program UPD). Like with the SPLIT program, if two events of the same type (L, R or D) have the same origin time to the second, one second is added to the file name part indicating seconds (see also section 13). The event will also be in chronological order in the CAT database.

*****VERY IMPORTANT *****

By default, the ID and file name are not changed by using the UPDATE program. However, optionally
this can be done. The user gets the question:

Keep sfile name and ID (default=enter) or synchronize sfile name and ID with origin time (s)

By synchronizing, the S-filename and ID is synchronized to the origin time. If no origin time, the earliest
time of the phases is used. At the same time an S is put in on the ID line column 76 to indicate event
has been synchronized (this was L in version 11 and before). The S will remain for all future updates so it
only indicates that the event has been synchronized at some time. To find out when, see the history
of the ID lines. The S is only used for information. When the file has been synchronized, the action is set
to UPS.

This is VERY important in case data is taken out of the database for some special analysis and then put
back in to overwrite the original data. If the ID is the same, the correct event will be replaced.

NOTE: When an update takes place, the old location, magnitudes (except 3. if a different agency from
the default agency), residuals etc are removed. If an event cannot be located, the old location etc is lost.
This is intentional since the updated database should represent the data available. If a location should
be retained, special flags must be set, see section “Fixing location” (a '*' in column 45 in header line).

In order to keep track of how and when the database has been updated, every run of UPDATE creates
a log file of the update process. This file is located in a subdirectory of the database directory (default
BER). If e.g. updating REA, the logfiles will be in ../REA/BER_/LOG/ (unix). Filenames are similar
to S-files. Below is an example of a logfile with name 01-0000-00L.S199606:

1996 06 kk 99-09-08 14:30 03-1955-35D. 25-0337-29L. 6

The content is as follows: date and time of file updated, operator ID, time of update, event id of first and
last event of the month, number of events for month. The example above shows that June 96 has been
updated 2 times, the last time on September 10, 1999. For each update, one line is added to the top of
the file, so the update history is saved.

Note: If the command UPDATE(U) is used from EEV, only one S-file is updated (name stays the same),
and a general update should be made.
UPDATE recalculate moments if distances (or depths) change, however it does not change the Vp or Vs
velocities used if a change is made in MULPLT.DEF.

Problem: If UPDATE crash, there will not be a correspondence between S-files and the CAT data base:
Redo UPDATE.

UPD

The command UPD is very similar to the UPDATE command, however there is no modification of the
S-file except the ID line. The program can optionally synchronize the S-filename and ID with the origin
time in the same ways as UPDATE. The only difference is that the action is US instead of UPS to indicate
that no relocation has been made. The program is used to simply move single S-files into the monthly
CAT-files without relocating. It is mainly used to manipulate database events already processed. E. g.
if ISC data a available and it is desirable to have it in individual files to be able to use EEV, the
same data can then be copied into the CAT part of the database using UPD without modifying the
original solutions. The data must be in the CAT part of the database in order for S ELECT to work fast.
KNOWN BUG: On Sun OS, it seems that UPD can only operate on up to a 4 year time period.
Chapter 15

Using filenr.lis, DIRF and DELF

DIRF

The DIRF command is a useful program for making a file with a numbered list of files from a directory. The command makes a file with file name filenr.lis e.g. when working with many waveform files with long names, a DIRF is first made, and subsequent programs then get file names from filenr.lis, either by using the whole list, or just a given number. This is handled with routine filename (in LIB). Below are some examples of using DIRF with SEISAN data files.

```
dirf 9101-10*
  # 1 9101-10-0915-15S.KMY_01
  # 2 9101-10-1510-55S.N2F_08
  # 3 9101-10-2333-44S.N3F_06

dirf 9101-10-0915-15S.KMY_01 9101-10-2333-44S.N3F_06
  # 1 9101-10-0915-15S.KMY_01
  # 2 9101-10-2333-44S.N3F_06
```

The wildcard '*' above indicates that all files from the 10'th is wanted. Many programs use the same subroutine to get the file name from filenr.lis. This means that most programs using filenr.lis assume that if a name given is less than or equal to 4 characters, it is a number so file names less than 5 characters cannot be used when the program asks for “Filename or number”. For a very long listing it might be an advantage to only get the first or the last 20 files and dirf has the corresponding options -head and -tail respectively. This argument must be the first argument like dirf -tail *.txt.

DIRF is dimensioned to a maximum of 99999 files.

DELF

DELF is a simple program that allows the user to delete a file that is listed in a filenr.lis file or another index file. First run DIRF to list the files that you want to delete. Then start DELF and choose the number of the file to delete, '?' shows the contents of filenr.lis. In addition, DELF also has an option to delete all the files in the filenr.lis or index file. This is a useful option if selected files in a data has to be deleted. If e.g. all S-files from a particular agency has to be removed, run SELECT first.
and then DELF.
Chapter 16

Making a bulletin, BUL

The bulletin program BUL is intended for writing seismic bulletins in a nice format. The output is written to a PostScript file.

Input files:

1. A monthly data file: This file can be made by COLLECT or SELECT

2. BUL.INP: This file must be in DAT or in the local directory. In this file the layout of the front pages are decided, as well as the font selection for the main bulletin. There are ample comments in the file on how the commands are written.

Some special format features:

Type 3 line: If the first 5 columns in a type 3 line are: " Bul: ", then the rest of the line is interpreted as text line that is written in the bulletin. In this way comments to certain earthquakes can be written into the bulletin. Type 2 line: Maximum intensity and casualty/damage reports are included in the bulletin if found in the S-file.

How to run the program:

Type bul -h this gives you a list of the different options like this:

Options:

-frontpage: Only frontages are printed.
-nofrontpage: No frontages are printed.
-onlyhypo: Only hypocenter solutions are printed.
-minmag x.x : Only hypocenter solutions with magnitude than the requested are printed.

The last option may be used in cases where the number of earthquakes is very high, so that it is preferable to report phases only for events above a given magnitude.

You can also run the program without any options, in which case the default values used are:

i) All phases are reported.

ii) Front pages are printed.

You will always be asked for the name of the S-file.

247
Output file:
The output file is called \texttt{bul.ps} and is a PostScript file that you can print. Optionally, a limited number of pages can be selected from the \texttt{bul.ps} file for printing. The header page is still included and the page numbers correspond to the original page numbers.
Chapter 17

Reports and statistics

SEISAN has several programs for extracting and writing out data for plotting or printing statistics, most of which will be listed in this section.

17.1 REPORT, extract data from CAT file

The program extracts parameter data from all header lines in a CAT file and rearranges the data in a table. In additions, there is an option to rearrange order and location of magnitudes on the header line. Below is an example of a run where the input CAT file is called collect.out:

report collect.out

Below is shown parameters which can be chosen for output. A return will chose all, placing any character under a field will chose that parameter in the output. Each field starts with a capital letter and ends within the following blank. The order of the output can be changed by placing a number under the field and fields will be written out in the order of the numbers. E after time, lat, lon and dep are errors, L E is distance and event id s, F is both fix flags and A is agency for magnitude. The following example shows that Mc, Depth(Dep) and Time with error are selected and written out in given order.

Date TimeE L E LatE LonE Dep E F Aga Nsta Rms Gap McA MlA MbA MsA MwA Fp Spec
30 45 20 10

Date TimeE L E LatE LonE Dep E F Aga Nsta Rms Gap McA MlA MbA MsA MwA Fp Spec
x x x x x x x x

Number of output fields 8
Number of events 12
Number of events with spectra: 1
Number of events with fault plane solution: 3
Number of events with error estimates: 16
Number of events with mc : 10
Number of events with ml : 10
Number of events with mb : 8
Number of events with ms : 5
Number of events with mw : 12

Output report file is report.out
Output nordic file is report_n.out
Output of choices used in report.inp

The report.inp is a file with the choices used. Report can use that file (or a file with the same format and a different name) as second argument:

report collect.out report.inp

in order to use a fixed set of choices.

Content of report.out

<table>
<thead>
<tr>
<th>Year</th>
<th>Date</th>
<th>Latitud</th>
<th>Longitud</th>
<th>Depth</th>
<th>NST</th>
<th>GAP</th>
<th>Ml</th>
<th>STRIK</th>
<th>DIP</th>
<th>RAKE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1996</td>
<td>6 3</td>
<td>47.776</td>
<td>153.222</td>
<td>0.1</td>
<td>12</td>
<td>348</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>6 6</td>
<td>62.652</td>
<td>4 940</td>
<td>15.0</td>
<td>13</td>
<td>270</td>
<td>2.9</td>
<td></td>
<td>28</td>
<td>61</td>
</tr>
<tr>
<td>1996</td>
<td>6 6</td>
<td>62.634</td>
<td>5 068</td>
<td>15.0</td>
<td>13</td>
<td></td>
<td>2.9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>6 6</td>
<td>62.652</td>
<td>4 940</td>
<td>15.0</td>
<td>13</td>
<td>270</td>
<td>2.9</td>
<td></td>
<td>28</td>
<td>61</td>
</tr>
<tr>
<td>1996</td>
<td>6 7</td>
<td>59.841</td>
<td>5 127</td>
<td>12.0</td>
<td>12</td>
<td></td>
<td>1.9</td>
<td></td>
<td>8</td>
<td>41</td>
</tr>
<tr>
<td>1996</td>
<td>610</td>
<td>-13.481</td>
<td>167.130</td>
<td>200.1</td>
<td>301</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>625</td>
<td>61.656</td>
<td>3 363</td>
<td>14.9</td>
<td>35</td>
<td></td>
<td>3.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>7 5</td>
<td>61.282</td>
<td>4 825</td>
<td>7.1</td>
<td>10</td>
<td></td>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>713</td>
<td>61.416</td>
<td>3 870</td>
<td>12.1</td>
<td>9</td>
<td></td>
<td>1.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>718</td>
<td>60.156</td>
<td>2 070</td>
<td>15.0</td>
<td>9</td>
<td></td>
<td>1.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>718</td>
<td>51.438</td>
<td>157.737</td>
<td>29.9</td>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>726</td>
<td>61.755</td>
<td>2 293</td>
<td>22.1</td>
<td>9</td>
<td></td>
<td>1.8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The file report_n.out contains the input data with the only difference that the magnitudes have been moved around on the header line. This can be practical for later plotting with EPIMAP. If no magnitude selection has been made, the magnitudes will come in the order Mc, Ml and Mb. If no magnitude of that type is available, the output field is blank. The magnitude selected is the first to occur of the corresponding type. If other magnitudes are to be selected, numbers can be used to select any 3 magnitudes in any order. If it is important to select magnitudes by agency also, use program MAG. REPORT can also give a numbered output by adding the second or third argument -n.

17.2 NORHEAD, making a compact Nordic file from a Nordic file

You must give arguments: First is input file, optional second is output file, if an optional second or third argument is -mag, magnitudes from following header lines are moved up to empty magnitude spaces on first line. The program was earlier called COMPACT (version 7.2 and earlier).
17.3 STATIS, statistics of databases

This is a simple program for making statistics of stations used in the database or in a file. The program will ask the following questions:

1. Information about which stations should be searched for in the database. There are several options for entry:
   a: Give a filename with the stations listed one per line. The format is .a5. The file name MUST have a '.' not to be confused with option (b) below.
   b: Give stations, one per line, enter to finish, enter for def file statis.def
   c: Just make a return and the stations given in file statis.def will be used. The file has one station per line and can be located in either the working directory or DAT.

2. Standard questions about base or filename and time interval

3. Question about counting all phases. This means counting the occurrence of a station for each phase for that particular station. This can give the total number of phases read at a particular station in a given time interval which can be more than the number of events. If not counting all phases, the program gives the number of events recorded at the station.

The output from the program could be as follows:

<table>
<thead>
<tr>
<th>Station</th>
<th>Local Ev.</th>
<th>Local S.</th>
<th>Distant E</th>
<th>Distant S</th>
</tr>
</thead>
<tbody>
<tr>
<td>KONO</td>
<td>0</td>
<td>21</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>KMY</td>
<td>24</td>
<td>10</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>ODD</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EGD</td>
<td>28</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>ASK</td>
<td>29</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>HYA</td>
<td>16</td>
<td>1</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>SUE</td>
<td>16</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>FOO</td>
<td>18</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>NRA0</td>
<td>86</td>
<td>9</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>MOL</td>
<td>38</td>
<td>17</td>
<td>14</td>
<td>1</td>
</tr>
<tr>
<td>NSS</td>
<td>9</td>
<td>6</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>MOR</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOF</td>
<td>25</td>
<td>13</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>TR0</td>
<td>12</td>
<td>2</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>BJO</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KBS</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>JMI</td>
<td>16</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KTK1</td>
<td>22</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>ARA0</td>
<td>66</td>
<td>6</td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

NET NWAW
NSN 147
JMI 21
KNN 10
W_L 2
252  

252  CHAPTER 17. REPORTS AND STATISTICS

W_E  1
W_S  2

Number of events selected with given stations  222
Number of events selected with more than ---  98
Number of events with no phases  0
Number of events with waveforms  168
Number of events with only waveforms  0
Number of events with 2 or more waveforms  9
Total number of waveform files  183
Total number of local events  200
Total number of regional events  0
Total number of distant events  34
Total number of events  234
Total number of records  2830

Output files are: statis.out
                statab.out (station statistics only)

The top part shows the event statistics by station. Local Ev is number of local events (readings if so
specified above) (type L and R) at the station, Local S means number of local events ONLY recorded at
that station, Distant E and distant S is the same for distant events (type D). The middle parts shows
the number of waveform files NWAV from different networks NET as indicated by the first 3 letters of
the waveform file name after the "." At the bottom is a summary statistics most of which should be
self-explanatory. The information about ".. more than given stations" means that in addition to the
stations searched for, the event had additional stations not used in the statistics.

17.4 CATSTAT, statistics of catalogs

This program calculates the yearly, monthly and daily number of events from a given earthquake catalogue
and plots the results (written by Mario Villagrán). The input is a standard Nordic file containing only
the header lines (compact file). The output is given in three different files with following default file
names:

catyear.out :  Output catalogue of the yearly number of events.
This file contains two columns of data corresponding
to year and the number of events.
catmonth.out :  Output catalogue of the monthly number of events.
This file contains three columns of data, correspond-
ing to the year, month and the number of events,
respectively.
catday.out :  Output catalogue of the daily number of events. This
file contains four columns of data corresponding to
the year, month, day and the number of events, re-
spectively.
cathour.out :  Hourly distribution of events within a day interval.

In addition, a series of files with gmt in name give similar output for use with gmtxy (only Unix). The
output files can then be used for plotting the histograms for the desired time interval at yearly, monthly
17.5. **SWARM, FINDING EARTHQUAKE SWARMS**

or daily intervals. If desired, the corresponding histograms can be plotted interactively on the screen or can be printed. Several other routine programs such as grapher, xyplot, gnuplot or GMT, etc., can also be used for this purpose. The general purpose of this program is to evaluate the catalogue completeness. When run for different magnitude intervals, one can detect the magnitude thresholds above which the catalogue can be considered complete.

### 17.5 SWARM, finding earthquake swarms

The program is used to identify seismic swarms in a catalog. Input to the program is a CAT file with many events and some manually entered parameters. Output is identified swarms. The output file swarm.out contains all swarms organized as 'events'. In the header line is given the center for area identified and the 'magnitude' is the number of events in the area divided by 10. The rest of the line is information from first event in swarm.

**Principle of selection:**

The area is divided into a lat-lon grid. Around each grid point, there is a cell with radius small. The program first checks the number of events in each cell for the whole catalog. It then checks each cell to find which has more than the minimum number of events to constitute a swarm under the condition that enough events are within the required time window. For each time window, with enough events, a swarm is declared so a swarm lasting e.g. twice the time window will be declared as two swarms. An additional condition is that the number of events is larger than the normalized background activity. The normalized activity is calculated as the activity in the large cell normalized for area to the small cell, and normalized in time to the window for the swarm.

### 17.6 STATSTAT, number of events per seismic station in catalog

The program reads Nordic file input data and writes out text files giving the number of events per station.

### 17.7 LSQ, plotting a linear relation or a curve

A simple program to make and plot a least squares or maximum likelihood relation between two parameters, plot a curve using the xy parameters or plot a histogram using the xy parameters. Input is from a file with two columns x and y. If a linear fit is made, the program also makes an output used with GMT in order to make nice plots. The PostScript output file is `lsq.eps` and the GMT file is `lsq_gmt.out`. In order to produce the GMT plot (only Unix), use command `gmtxy lsq_gmt.out`. Plot a curve. The curve can be plotted with points only, lines joining the points with points plotted or just lines between points. X and y-axis for the curve plotting can be linear or logarithmic.

For plotting histograms, the input data must be equally spaced in x like e.g.

```
0 4
10 22
20 11
```
Examples of plots are found under program GETSTRESSDROP
Chapter 18

Waveform file management tools

This section describes the programs used for modifying and checking waveform files. The most important features are to add or subtract channels and modify headers. A special program in this group is GET_WAV which checks data bases for availability of waveform files. New from version 7.1 is that SEISAN also can handle other waveform formats, however not all programs can work with all formats. This will be indicated with each program. The following programs are available:

- **APPEND**: Append two or more waveform files following each other in time
- **AUTOREG**: Automatically register events
- **CONGAP**: Check completeness of continuous waveform database
- **CONNOI and EVANOI**: Compute noise power spectral density and evaluate out to produce plots
- **DATABASE2MSEED**: Convert waveform data to miniseed based on parametric database
- **GET_WAV**: Check for available waveform files
- **GET_ARC**: Extract wav files from archive and register in S-file
- **MSCUT**: Cuts MiniSEED files into 1 hr or shorter files
- **RDSEED_MANY**: Simple way to chop up a seed volume
- **RESAMP**: Resample waveform files
- **SEIASC**: Convert SEISAN waveform files between ASCII and binary form
- **SEICUT**: Extract an interval of a waveform file
- **SEIDEL**: Splitting up a SEISAN waveform file in 2
- **SEISEI**: Split and merge SEISAN, GSE, SAC and MiniSEED waveform files
- **SELSEI**: Find waveform files with given stations
- **P_ALIGN**: Time shifting waveform data to align P or S-phase arrival times
- **WAVETOOL**: Extract waveform data
- **WAVFIX**: Fix waveform file header time correction, make standard file names, change headers etc.
- **WAVFULLNAME**: Print full file name including path for waveform file.
18.1 APPEND, Append two or more waveform files

The program uses a *filenr.lis* input file. All files are read, and then written out as one new file. The maximum number of channels is `max_chan_out` which is set as a parameter (currently 7). Only the first `max_chan_out` channels are used or less if fewer channels in file. A blank line followed by a new group of files will make a new output file. There is also a question of how many files should be merged at a time so e.g. every 10 files will be merged. This is independent of the option with blank lines. The output file cannot have more than standard SEISAN dimension number of samples (more than 1 200 000, see file `../INC/seidim.inc` for exact number) per channel.

It is assumed that all channels have the same sample rate and that files follow each other in time.

Input format can be many but output format is SEISAN.

**AUTOREG, automatic registering of events**

When a large number of waveform files are available and it is known that they are real events, it might be an advantage to automatically register them into a database. Remember, the database can be made beforehand with MAKEREAA. If the filename follows the SEISAN filename convention, the date and time used to generate the S-file are taken from the filename. Otherwise, the file is read to get the date and time from waveform headers. Obviously, the first option is faster. It is possible to register events both to the default database, any other database or the local directory. To run the program, make a *filenr.lis* of the waveform files and run AUTOREG. It is possible to put blank lines into the *filenr.lis* to separate into events, in case there is more than one waveform file from the same event. All waveform files before a blank line are put together into one S-file. Optionally, the waveform files can also be moved or copied to WAV or a WAV database subdirectory (including year and month). This can either be the default parameter `COPY_WAV_DIR` (in *SEISAN.DEF*) if different from blank. `COPY_WAV_DIR` should be the same as the database used by the S-files. However an optionally data base directory entered interactively can also be used.

You can run AUTOREG with the flag `-model`, to set the location model indicator in the header line, e.g. `autoreg -model G` for the G model. Then you get the questions:

```
autoreg
Event type for all events: Local: L (default)
                  Regional: R
                  Distant: D

Move (m) or copy (c) waveform files to WAV (enter=m) ? m
Files will be moved to default WAV base: TEST_
Enter new WAV data base to copy or move to enter for default base or m for main WAV directory
LVC

Operator, max 4 chars
jh
/home/s2000/seismo/REA/LVC__/2010/01/18-1226-15L.S201001
File transferred to WAV base LVC__ **********
2010.018.12.26.52.0695.IU.LVC.10.LHZ.R.SAC
......
```
In this case waveform files were moved to data base LVC instead of the default TEST.

Now comes a listing of waveform file names and S-file names. The program will check if the event is already registered and the same options are available as in program SPLIT (section 13). Since AUTOREG automatically creates S-files for all events in filenr.lis, they will all be given an event type.

**CONGAP, check completeness of continuous waveform database**

This program checks for completeness of continuous data for a given time interval. The program reads the waveform data to see what data are available and checks for gaps, defined by a constant amplitude value (e.g. 0). The input can come either from an input file (`congap.par`) or the command line.

Parameters in the input file are:

- **CONT BASE**: name of database, you can have more than one
- **START DATE**: start time and date of interval to be read (yyyymmddhhmmss)
- **STOP DATE**: stop time and date of interval to be read (yyyymmddhhmmss)
- **INTERVAL**: duration of intervals read at a time in minutes (e.g. 60. for one hour)

When started from the command line, the same parameters can be given:

```bash
congap -start <yyyymmddhhmmss> -stop <yyyymmddhhmmss> -cbase <text> -interval <number>
```

The output file (`congap.out`) looks like this:

```
EDI HHZ 20080101 0000 0.00 3600.00 3600.00
EDI HHN 20080101 0000 0.00 3600.00 3600.00
EDI HHE 20080101 0000 0.00 3600.00 3600.00
EDI HHZ 20080101 0100 0.00 3600.00 3600.00
EDI HHN 20080101 0100 0.00 3600.00 3600.00
EDI HHE 20080101 0100 0.00 3600.00 3600.00
...```

The fields are station and component code, date and time, expected duration and actual time with data. The output file can be used to produce plots showing data completeness (tool for this not included). When the program runs it also produces a summary output at the end, where the last column gives the percentage of data completeness, and the actual and expected times are in seconds:

```
--------------------------------------------
#   stat comp    actual    expected    %
--------------------------------------------
 1 EDI  HHZ  86400.00  86400.00  100.0
 2 EDI  HHN  86400.00  86400.00  100.0
 3 EDI  HHE  86400.00  86400.00  100.0
--------------------------------------------
```

**CONNOI and EVANOI, noise power spectral density**

These two programs with the help of GMT allow to produce noise power spectral density (PSD) plots similar to the ones produced by the PQLX software. CONNOI is the tool that reads the continuous
database and produces output files that are evaluated by EVANOI. The computation of the noise PSD follows the method described by McNamara and Buland [2004].

To run CONNOI on a SEISAN continuous database BER use for example:

`connoi -start 20100501 -stop 20100502 -cbase BER`

To run CONNOI on an archive structure BER use for example:

`connoi -start 20100501 -stop 20100502 -arc BER`

In both examples BER is the database, you can also specify 'def' and the program will take all default continuous databases or archives defined in SEISAN.DEF. The default output filename is connoi.out.

Example of output:

```
stat comp date and time duration frequency noise PSD
-------------------------------
BER HHZ 20100501 0000 0.00 3600.00 0.00200 -159.14
BER HHZ 20100501 0000 0.00 3600.00 0.00204 -159.14
BER HHZ 20100501 0000 0.00 3600.00 0.00209 -159.14
...```

The output from CONNOI can then be used as input to EVANOI. You can enter station and component, give a time interval, select a time of day interval, and chose a reference station. EVANOI produces GMT plotting scripts files that are named after the station. Then simply run the script file to get a plot.

You can also output the data in a compact format using the -compact option. This output does not work with EVANOI, but it is fairly easy to read in your own scripts. An example output:

```
CONNOI: NF= 500 START=2008120101 STOP=2008120103 DURATION= 60.0min, SPACING= 50.%
STATN CMP yyyyMMdd HHHMMSS.FF SPECLEN -2.69897 -2.68955 -2.68014 ...
BER HHZ 20100501 0100 0.00 3600.00 -102.94 -102.94 -102.94 ...
BER HHE 20100501 0100 0.00 3600.00 -115.81 -115.81 -115.81 ...
VASAV BHZ 20100501 0100 0.00 3600.00 -134.16 -134.16 -134.16 ...
VASAV BHE 20100501 0100 0.00 3600.00 -124.60 -124.60 -124.60 ...
...```

By default, CONNOI calculates spectra every 30 minutes, with each spectra calculated over a 60-minute interval. You can change this using the -spectlen and -spectspace options, where -spectlen gives the interval over which to calculate the spectra (in minutes) and -spectspace gives the spacing between spectra with respect to -spectlen (1.0 for no overlap, 0.5 for 50% overlap, etc). For example:

`connoi -start 20100501 -stop 20100502 -cbase BER -spectlen 60 -spectspace 1.`

will calculate spectra over a 60-minute interval, spaced every 60 minutes (no overlap). Finally, you can calculate the spectra for a single channel of a single station using the -sgram option. This option always uses the -compact output format and sets spectspace to 1.0. For example:
Figure 18.1: Example of plot created using CONNOI and EVANOI. The plot shows noise PDFs for station BER for three components in January 2015.
connoi -start 20100501 -stop 20100502 -cbase BER -sgram TOTI BHZ

**DATABASE2MSEED, convert database to miniseed**

This program can be used to convert the waveform data that is linked to from a parametric database to miniseed format. The user is asked to enter the database name, and start and end time for the operation. One should do a small test with a copy of parts of the database before running it through the complete database.

**WAVETOOL, extract and convert waveform data**

The program was called extract in SEISAN 7.2 and earlier. The program extracts all or selected time sections of waveform data, optionally applies some signal processing and then creates the output file(s). The input formats supported are SEISAN, SEED, MINISEED, GSE, HELMBERGER (only made by MULPLT), GURALP (one channel files) and SAC (all platforms) and output formats are SEISAN, MINISEED, GSE and SAC (all platforms) and SEED on Linux. The program can be used as a conversion program between these formats, instead of using e.g. SACSEI. It would also be possible to convert for example SAC and GSE files to SEISAN in one go. When creating GSE, MINISEED or SAC files, the respective format code is added to the filenames. The program can also extract data from a SEISAN continuous data base or a large SEED file.

File name convention for output files:

In interactive mode, the output file name is SEISAN by default since there is no question about file name.

No output file name given: wavetool.out for SEISAN output file

wavetool.out_SEED for SEED output file

wavetool.out_MSEED for MSEED output file

wavetool.out_GSE for GSE output file

Output file name given as SEISAN: SEISAN name for SEISAN output file SEISAN_name_SEED for SEED output file SEISAN_name_MSEED for MSEED output file SEISAN_name_GSE for GSE output file

No output file name given and output format is SAC:

SEISAN_name_component_SAC

Output name given: That name is used exactly as specified. There is no check if file exists and there is no ending indicating format

Output of one channel files with SEISAN names: The channel code is added to avoid overwrite of different channels from the same station

Output format is SEISAN and output file exists: _SEI is added to file name

Note that for MINISEED writing, only Steim1 compression is used. Integer format is also possible, but requires a parameter change in WAVETOOL and recompilation.

There are two input options: (1) a single S-file or a list of S-files created with DIRF (is also an index
file), which points to the waveform data or a `filenr.lis` type of file that gives the waveform file names; (2) a waveform file or a list of waveform files. The program can be started either interactively, without arguments, or non-interactive by specifying the commands as arguments.

Filtering: WAVETOOL only supports band pass filters hardwired to 4 poles and one pass (forwards). So if WAVETOOL is started from MULPLT to extract data as seen on the screen, the filter constants will ONLY be passed to WAVETOOL if 4 poles and band pass.

The arguments are: `-sfile <sfile-name>`: The S-file name of the event you want to extract waveform data from.

-`wav_files <file-name>`: Extract from a list of waveform files in `filenr.lis` format. Input from S-file will be ignored. This option merges all files from list (if within a 'reasonable time window') and program is then partly doing what SEISEI is doing. If file-name is given as ‘SEISAN’, then the output file name follows SEISAN convention.

-`wav_in_file`: input waveform file.

-`maxpoints`: Skip points to get approximately maxpoints

-`wav_out_file <file-name>`: Name of waveform output file, not used if output format is SAC.

-`start <time>`: Start time can be used to set start time the same for all channels instead of using chansel file, time can be absolute or relative to beginning of the first trace. Options are Start time relative earliest channel, Abs start time string yy...s.sss for all channels. If ABS time used, string MUST contain a ‘.’. Abs start time string yyyyymmdhhmmss (integer), used to define cont start time.

-`duration <time>`: Select duration of time window in seconds if `-start` is used.

-`interactive`: enable interactive mode

-`cbase`: name of file with selected continuous databases, all bases is default

-`command_file`: give arguments to wavetool in a file rather than on the command line (used by Seisweb)

-`cwav`: Input from the SEISAN continuous data base, useful for extracting intervals

-`cseed`: Input from large SEED file, similar to `-cwav -wav_in_file <file-name>`: Input of one waveform file

-`arc`: Input from BUD or SeisComp archive.

-`sfile <file-name>`: Input of S-file name

-`format <output format>`: The output formats supported are SEISAN, SAC, GSE (GSECM6) and GSEINT. In case of SEISAN or GSE, multi-trace files including all selected traces are created, while for SAC, single trace files are generated. SEED is also possible on Linux.

-`chansel <file-name>`: Input file to select channels and time windows. The first line contains number of channels. The following lines give station code, start time (both absolute and relative to earliest trace allowed) and duration. If start time and duration are set to 0, complete traces are selected.

-`chan_out_file`: file name of file with channel description if this option is given, program terminates after writing out the file.

-`chandef <file-name>`: change station and/or channel names following standard also used in the conversion programs, for example:

  `Header line text (29 chars) ... NetCd (5 Chars), comment in next line`

  `chan stati  comi stato  como, In and Out definitions`
-ground : compute displacement, velocity or acceleration (1,2,3)
-seed_location: seed location code NOT USED.
-rsam: compute 1 min rsam data.

Example

Select the first three complete traces
3
1 0 0.0
2 0 0.0
3 0 0.0

Use absolute start time, and duration of 180 seconds
3
1 19991001124500.000 180
2 19991001124500.000 180
3 19991001124500.000 180

Use relative start time of 60 seconds from beginning of earliest trace, and 300 seconds duration
3
1 60 300
2 60 300
3 60 300

The program assumes that a large number is absolute time.

-chan_out_file <file-name> : Name of text file containing a list of available channels from a list of waveform files. If wav_out_file is not specified, program terminates after creating the list.

Example

1 KBS BV Z 1996 6 3 20 2 18.991 6000 20.000 2666.400 299.950
2 LOF S Z 1996 6 3 20 5 5.531 5800 50.000 2832.940 115.980
3 MOL S Z 1996 6 3 20 5 24.984 10000 50.000 2852.393 199.980
4 FOO S Z 1996 6 3 20 5 34.156 9650 50.000 2861.565 192.980
5 HYA S Z 1996 6 3 20 5 36.078 9900 50.000 2863.487 197.980

-filter <flow> <fhigh> : bandpass filter limits

If a filter is used, a 4 pole Butterworth filter is used one way. No other filters are possible. Filter in WAVETOOL

-ground <1,2,3> : compute displacement, velocity or acceleration (1,2,3)
-ichan <id> : select one channel only

-interactive : Flag to specify non-interactive use, in which case the program does not ask any questions (for example given by SEISAN-autodrm interface), default is interactive.

-seisweb : Flag to indicate that the program is started by SEISWEB.
18.2 GET_ARC, EXTRACT WAVEFORM FILES FROM THE ARCHIVE CORRESPONDING TO S-FILES AND REGISTER THE NAME IN THE S-FILE.

- **maxpoints <number>**: Number specifies the total number of points desired for the total time window covering all selected traces. This option is meant to reduce the number of points to what is needed to visualize correctly plot the traces. When plotting the trace using a number of pixels, which is smaller than the number of points on the trace, samples are plotted on top of each other for one time sample. This results in the maximum and minimum being plotted at the same place on the x-axis. The idea now is to reduce the trace to these maxima and minima only. Then using twice the number of samples than pixels will allow to visually correctly show the trace. Note that this is not a resample routine. Option mainly used with SEISWEB.

- **stat_out**: write out station location file, simple xy output file (station_list.out)

- **rsam**: convert data to RSAM (1-minute absolute average), which is commonly used in volcano seismology

- **resp_out**: write out list of all response files for channels given in waveform files (respfile_list.out)

Interactive input

The interactive input has less options as the mon interactive input, however the option are as above. The questions are:

- Filename of s-file or waveform file, number or filenr.lis
- Maximum number of points in output trace, return for all same as -maxpoints
- Ground motion output (dis = 1, vel = 2, acc = 3, none = return)
- Filter low and high, return for no filter
- Select all data, y=return, n
- If answer is n, then the following 2 questions come: Number of channels to read, Channel number, start time and window

Output formats (SEISAN, GSE (def=CM6), GSEINT, SAC, MSEED) Default is SEISAN=return

Another way of extracting waveform data is using MULPLT where many traces can be extracted as a binary SEISAN file (using WAVETOOL in the background) or a single trace as an ASCII file.

**Accuracy of extracted data**

If the data is filtered or corrected for instrument response, the number out can be less than one and an output file of zeros can be made. If the output format is SEISAN, the values will always be scaled to avoid this and the appropriate scaling factor is included in the waveform file. Subsequently reading of these files in SEISAN will produce the correct values. For this reason, it is advised to use SEISAN as output format when filtering or correcting for instrument response. SAC data can have values less than 1.0 so only if written in SAC or SEISAN will the values be correctly represented. If a more specific combination of filters and and response removal is required (like making a Wood Anderson trace with an additional filter) it can only be done for one event using the MULPLT OutW option which creates an ASCII Helmerger file which then can be converted with WAVETOOL to any other format. SAC input data is checked for max values. If smaller then 10, output will be scaled in SEISAN format. Note: Parameter MERGE,WAVFORM in SEISAN.DEF sets the network extension of extracted files.

**18.2 GET_ARC, extract waveform files from the archive corresponding to S-files and register the name in the S-file.**

The program uses an S-file as input (one or many files). The stations to select from the archive can be user specified, all stations with readings, all stations in archive or all stations in a given distance from the epicenter. All channels for each station are automatically selected and there is no option for selecting e.g. all Z-channels. The output file get_arc.out contains the input data with the waveform file line added. If the process is done twice and the same waveform file name is created, it will only be recorded once in
the S-file. The output file must be split into the data base and possibly overwrite existing S-files. The program can also be run with arguments for the operator and file e.g. `get_arc collect.out bb`. This option is used when the program is called from EEV with command ARX.

Example run

```
get_arc
Give operator
jh
Give input file
collect.out
Interval in number of seconds before and after origin time
Default (enter) is 30 and 300
50 500
Extract stations with readings: enter
  all channels in archive: a
  for given stations interactively: s
  for stations in a file: f
  for stations at given distance to epicenter: d
  for stations to given distance to given point: p
a
********** Event  1  2011 129 10 0 0.0 L
wavetool -start 20110129095910 -arc -duration 550.0 -wav_out_file SEISAN -cbase cbase.inp -format MSEED
Number of archive channels defined 3
Total duration: 551.050476
Output waveform file name is 2011-01-29-0959-09M.NSN___003
Extracted file 2011-01-29-0959-09M.NSN___003
End of s-file

Number of events in input file 1
Number of events skipped 0
Output file name is get_arc.out
```

In details, the options are: Default: Use stations with readings. If a station has more than one reading, it is only selected once.

- **a**: All channels in archive. If a very large network, this option might select too much.
- **s**: For given stations. Give station codes, one per line.
- **f**: For given stations. Give stations from a file, one per line.
- **d**: Use all stations in archive within a given distance. If there is no epicenter, the stations in S-file will be used. If no stations in S-file, all station in archive will be used.
- **p**: Use all stations at a given distance from a point. The user will be asked for the latitude and longitude of point and distance.
18.3  GET_WAV, GET LISTING OF AVAILABLE WAVEFORM AND RESPONSE FILES

The program uses a CAT file as input and checks for availability of all waveform files listed. For each channel, there is a check on existence of corresponding response files. A typical run is shown below:

```
get_wav
    INPUT FILE NAME
    select.out
    Where to copy files to, default . on Linux and blank on Windows
```

```
Full path name : /net/seismo/users/jens/TD/WAV/1996-06-03-2002-18S.TEST__012
Full path name : /net/seismo/users/jens/TD/WAV/1996-06-03-1917-52S.TEST__002
Full path name : /net/seismo/users/jens/TD/WAV/1996-06-07-1324-51S.TEST__009
Full path name : /net/seismo/users/jens/TD/WAV/1996-06-10-0105-42S.TEST__014
Full path name : /net/seismo/users/jens/TD/WAV/1996-06-23-0059-47S.TEST__001
Full path name : /net/seismo/users/jens/TD/WAV/1996-06-25-0336-34S.TEST__032

75

Total number of events 6
Number of events without waveform files 0
Number of waveform files 8
Number of waveform files present 8
Number of waveform files missing 0
Number of cal files found 28
Maximum number of cal files missing 29
Output file with events is get_wav.out
Output file with waveform file names is copy_wav.out
Output file with cal files is copy_cal.out
Output file with waveform file names missing is copy_wav_missing.out
Output s-file with waveform file names missing is copy_wav_missing_sfile.out
Output file with missing calibration channels is copy_cal_missing.out
```

The ‘where to copy’ option can be used to write the complete path to where files are copied. This option is only used for waveform files, not calibration files. Note: On PC the files copy_wav and copy_cal have names copy_wav.bat and copy_cal.bat, respectively.

In the above example, a select.out was used. For each file, it is checked if the waveform and response files are available in the system. All waveform data bases and directories specified in SEISAN.DEF are searched. Calibration files are searched for in working directory and CAL. In order to extract the waveform files corresponding to the input CAT file, the output file copy_wav.out can be used to copy the files out of the data base to working directory. On Unix, just source the copy_wav.out file, on Windows, change the file to a .bat file (e.g. copy_get_wav.out wav.bat) and run it. For the calibration files there is similarly a file called copy_cal.out.

MSCUT chop up MiniSEED files

The program cuts up MiniSEED files into 1 hour, 15 minutes or x minutes files. Where 60 modulo x is equal to zero. Note that mscut is splitting miniseed files at block level and not at the first sample in
the new minute, in order to keep the original header information. To compile the program, the miniseed library `libmseed` by Chad Trabant is required. The options are

- `-H` Cut into one hour files (default)
- `-Q` Cut into 15 min files
- `-D` Cut into files with a duration defined by minutes
- `-V` Report program version
- `-h` Show this usage message
- `-p` Print details of header, multiple flags can be used
- `-s` Print a basic summary after processing a file
- `-r` Specify record length in bytes, required if no Blockette 1000
  file

18.4 RDSEED MANY, chop up seed file

The program reads a large SEED volume and divides it up into several files of the same size. It calls rdseed, so rdseed must be installed. Rdseed can do the same, but RDSEED MANY is simpler to use.

Example:

```
rdseed_many
Seed file name
test.seed
start time YYYY,MM,DD,HH,MM
2005 01 01 01 01
Interval in minutes
20
Number of intervals to extract
200
```

The output format is SAC, other format require a change of the program.

18.5 RESAMP, resampling waveform files

RESAMP is a simple resampling program, which can resample one or several waveform files. It only works with SEISAN format. All files are read, filtered and resampled. Then written out as one new file with the data from one or several input files. The maximum number of channels is `max_chan_out`, which is set as a parameter in the program, currently it is set to 7. Only the first `max_chan_out` channels are used or less if fewer channels in input file. It is assumed that all channels have the same sample rate and will be resampled to the same lower sample rate, which is an integer fraction of the original sample rate. If e.g. the original sample rate is 50, new rates of 25,10,5,2 etc can be obtained. The anti-alias filter is a single pass Butterworth with 6 poles. The user specifies manually both the decimation rate (2,5,10,25 in the above example) and the filter frequency. The new file(s) can have a new component specification, which is asked for interactively. Finally the user is asked for a new network code.

The input files(s) come from a `filenr.lis` file generated with DIRF. If more than one file is given in the `filenr.lis`, these will be put together in one file and some samples are saved from one file to the next in order to assure that there are no overlap problems when using the filter. IT IS ASSUMED THAT
ALL FILES HAVE THE SAME LENGTH OF TIME. The program will check if a following file has the correct header time based on the length of the previous file. If the following file starts before the end of the previous file (err_samp samples, default 70), it is assumed that the timing is wrong and that the files should follow each other. A warning is given and the program continues. If the following file has a header time that is more that a given err_samp samples after where it should be, it is assumed that the next file is missing and zeros are inserted in the channel data. The number of sample errors, err_samp, is hardwired in the program, currently 70. The program will continue to put data together in one file until there are no more file names in the filenr.lis file or a blank line is encountered. After a blank line in the input file, a new output file will be created. This can be used to make daily files of e.g. 2 weeks on continuous data by manually placing a blank line in the filenr.lis file for every 24 hours. The program recalculates the sample rate based on time in first and last file. The output file name will be given the standard waveform file name with type R for resampled like 1999-07-02-1112-22R.BERGE.005.

Works ONLY with SEISAN format

18.6 SEIASC, converting SEISAN waveform files to or from ASCII

A simple program to make an ASCII equivalent of a binary SEISAN file, or vice versa. It is the same call to use the program both ways. By using a filenr.lis file as input, many files are converted and the original filenames are kept with the addition of an A for ASCII or B for binary. If the files are converted back, the A or B is removed.

The program is useful for manually editing a waveform file or checking the content in case of problems. The program is also useful for moving binary files between different types of computer platforms (moved as ASCII files, not needed between platforms running SEISAN). Between PC, Sun, Linux and MacOSX, SEISAN programs will automatically adjust for differences in binary structure. The header format is exactly like the binary SEISAN files and the sample values are written in multicolumn format.

Works ONLY with SEISAN format

18.7 SEICUT, extract part of a waveform file

A simple program to extract out a section of a waveform file (any seisan primary format). A similar job can be done with wavetool. Syntax is:

seicut filename yyyymmddhmmss.s interval

The first sample to use is the first sample found before the start time, the output time interval (in seconds) will be the time from first to last sample, so if e.g. one second of data is asked for at a sample rate of 100 Hz, the time interval in header will be 0.99 sec and the number of samples output will be 100. If the interval is not available in any of the channels, the program will stop. The output file name will use a network code reflecting station code of first channel in input file and ‘CUT’ is added to the end of the file name. The same time window must be available in all channels.
18.8 SEIDEL, splitting a SEISAN binary file into 2 files

The program splits up waveform file into 2 files. Input can be file or list of files (filenr.lis created with DIRF). The questions are:

<table>
<thead>
<tr>
<th>Filename, ?, number or filenr.lis for all filenr.lis</th>
</tr>
</thead>
<tbody>
<tr>
<td>No of channels to remove</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>Channels to remove</td>
</tr>
<tr>
<td>1 3 6</td>
</tr>
</tbody>
</table>

The program will generate 2 new files, one with the channels removed and one with the remaining channels. The original file is still present.

Works ONLY with SEISAN format

18.9 SEISEI, splitting and merging SEISAN readable binary files

The program can merge several SEISAN, MiniSEED, SAC etc waveform files to one file or take one SEISAN, GSE or MiniSEED file and split it into single channel files. The program is intended for editing waveform files and merging files from different networks to one file. In order to use SEISEI for merging files, a DIRF must be made to make a filenr.lis file containing the files to be merged. The program will sequentially read filenr.lis and merge files which have start times within the time interval specified (3 minutes default). Once a gap of more than 3 minutes occur, a new output file is made. Merging to a new file can be forced by editing filenr.lis so the groups of files to be merged are separated by a blank line, however, within the group, the time difference can still only be the given time interval.

If two channels to be merged have the same station and channel codes and the same start time, the second occurrence will be ignored. If the station and channel codes are the same, but start time different, the user will be asked to confirm merging.

The program can also split up a multichannel file to files with only one channel. This can be used to remove unwanted channels by deleting selected channels and merging again. When the file is split up, the channel component is added to the file name. A filenr.lis file can also be used for splitting many files in one go. If a file is only to be split into only 2 files, it is more convenient to use the program SEIDEL (only works on SEISAN format), see above. SEISEI is also used in connection with MULPLT for merging files automatically based on waveform file names in an S-file.

Note: The network extension of merged files will be set by default to the value of parameter MERGE_WAVEFORM in SEISAN.DEF.

18.10 SELSEI, searching headers in SEISAN waveform files

A simple program to search headers in waveform files for files containing a particular station.

Works ONLY with SEISAN format
18.11  **P.ALIGN: Time shifting waveform data to align on P or S-phase arrival times.**

If one wishes to compare signals (align in time) from different earthquakes observed at the same station, the P.ALIGN program can be used. The program works by time shifting the waveform header times to a common time and then putting all the new waveform file names into an S-file. First use e.g. the SELECT program to extract information of earthquakes in a defined area that have been observed by a given station, and use GET_WAV to copy the waveform files to your working directory. Then execute P.ALIGN. The input is the Nordic file (e.g. select.out) and the station name for data to be compared. The output is:

- Waveform files with time shifted headers, all have the same time, but station names are labeled STA01, STA02 etc in the same order as given in input file. Only first 3 letters of station code is used.
- tsd.out : A file in Nordic format with the new waveform file names of the time shifted data. This file can be split and then used with EEV for plotting all traces. It can be split into a local data base or any other data base. Or copy tsd.out to a file e.g. named 27-1200-00L.S207011 and then start EEV.

The waveforms in the output file tsd.out can also be plotted with the command mulplt -sfile tsd.out.

The arrival time of the selected phase in the new waveform files is the pseudo date and time: 2070-11-27 12:00.

Note: The station name is renamed in the output. If there are more than one P-phase observation for a single event (e.g. Pn and Pg, or P read on two different channels) it is the first P that is time shifted. The program can only time shift 99 waveform files and they have to be in SEISAN format. The waveform file must be present in the working directory.

Figure 18.2 show an example.

18.12  **WAVFIX, fixing time correction and channel names in waveform file headers and make standard file names**

The input format can be any SEISAN readable format like SEISAN and MIniSeed. The output format is only MiniSeed.

It can easily happen that a waveform file has a wrong time in the headers, or that individual channels have wrong timing, for example introduced by different delays in the acquisition system that are not accounted for. WAVFIX can change all header times with a given constant time delay, or correct individual channels as specified in a parameter file (wavfix.tim). In addition, the file name will also be changed to reflect the header time change. Waveform file names were shorter in SEISAN version 6.0 so when using older files, the user might want to use standard file names. In cases where only channel names or timing of individual channels are changed, the filename can be kept the same. In this case a temporary file is created, which is later renamed to the original name.

WAVFIX can also change polarity. This is done by setting the output channel and station codes to the same as the input values in wavfix.def.
Figure 18.2: Example of aligning traces from 5 events for the same station. Note that the alignment is critically dependent on the original P-picks.
18.12. WAVFIX, FIXING TIME CORRECTION AND CHANNEL NAMES IN WAVEFORM FILE HEADERS AND MAKE STANDARD FILE NAMES

In case channel names are to be changed, this can also be done with WAVFIX. A definition file is needed for changing station, component or both. The parameter file name is wavfix.def and an example is given in DAT. For definition of the wavfix.def, see next section [19] on “Conversion programs definition file”.

WAVFIX can change header times and/or file names for one or many files. Before running the program, a list of file names must be made with DIRF. Below is an example where the header time is changed by 120 secs. No wavfix.def file is present (current or DAT directory).

To correct the timing of individual channels, you need to create the file wavfix.tim in either the DAT or working directory. WAVFIX checks if the file is present and applies the correction from the file as default. The format of this file is as follows:

Column 1-5: station code
Column 7-10: component code
Column 12-25: start date and time for time correction (can be empty)
Column 27-39: end date and time for time correction (can be empty)
Column 41:60: time correction to be added

NOTE: The file must have 3 header lines of any content before the station lines as shown below.

Example:

```
wavfix time correction applied to individual components
stat comp start time end time correction in
a5 a4 yyyyMMddhhmmss yyyyMMddhhmmss seconds f20.3
-----|----|--------------|--------------|--------------------
TEST SH Z 19500101120000 19600101120000 -0.015
```

File names of waveform files can be given to WAVFIX directly, from a filenr.lis file or from a Nordic format file. In case you choose the Nordic input, the waveform file names will be changed in the Nordic file (output file nordic.fix). This option is useful if you are correcting file names, since the entries in the S-files are otherwise not fixed.

ONLY the first waveform filename in the Nordic files is used.

WAVFIX will also take input from the prompt. Writing wavfix -help gives

```
wavfix -help
wavfix
usage: wavfix -infile <infile> [-uncertain_time -polarity] [-time_correction <x in seconds> ]
-uncertain_time add uncertain time flag to all channel headers
-polarity change polarity
-time_correction apply time correction of x seconds
```

The uncertain time option does not seem to work for MiniSeed output.

Example of running WAVFIX

```
No wavfix.def file, will use internal information for channel codes
This program will change header times in all headers
with the same amount. The waveform file name will be
```
CHAPTER 18. WAVEFORM FILE MANAGEMENT TOOLS

changed at the same time and adjusted to the standard name.
If no time correction is given, only the waveform names are adjusted.
In addition, channels names can be changed if a wavfix.def file is available

Time correction in seconds, return for no correction
Input options: (1) filenr.lis or waveform file name
              (2) Nordic file
Filename or number, filenr.lis for all
Input file name: 1994-06-16-1841-57S.TEST__019
Output file name 1994-06-16-1843-57S.______019

Input file name: 1994-10-04-1324-00S.TEST__016
Output file name 1994-10-04-1326-00S.______016

Input file name: 1994-10-04-1324-24S.TEST__016
Output file name 1994-10-04-1326-24S.______016

18.13 WAVFULLNAME

Prints full filename including path for a waveform file by searching directories and databases specified in SEISAN.DEF. Filename is to be given on prompt line, e.g. wavfullname 1996-06-13-1248-15S.NSN_003.
Chapter 19

File conversion and modification programs

There are mainly two types of files to convert, parameter files with readings and related parameters and binary waveform files.

PARAMETER FILES
CAT_AGA: Records the S-file header lines according to agency
EDRNOR: Converts USGS monthly bulletins (EDR files) to Nordic format
GIINOR: Converts from Geophysical Institute of Israel parameter format to Nordic
HARNOR: Converts standard Harvard CMT solutions to Nordic format
HYPNOR: Converts from Hypo71 readings files to Nordic format files
HINNOR: Similar to HYPNOR for Hypoinverse files, archive format
HSUMNOR: Converts from Hypo71 summary file format to SEISAN format
ISCNOR: Converts from ISC 96 column format to Nordic format
ISSCSV2NOR: Converts ISC csv format catalog to Nordic
ISCSTA: Converts ISC station list to SEISAN station list optionally selecting specific stations.
ISFINOR: Converts between ISF1.0 and Nordic
NORMSNOR: Converts from IMS, GSE and ISF to Nordic
KINNOR: Converts from Kinemetrics to NORDIC
NORCSV: Converts from NORDIC to CSV format using MAGNITUDE_ORDER given in SEISAN.DEF to setup priority of magnitudes.
NORGSE: Converts between Nordic format and GSE parametric format
NORHIN: Converts from Nordic format to Hypoinverse archive format
NORIMS: Converts from Nordic to and from IMS1.0
NORHYP: Converts from Nordic to HYPO71 format
NORQML: Converts NORDIC to and from QuakeML
PDENOR: Converts a PDE bulletin file to NORDIC format
RSANOR: Converts Andalucian Seismic Network data to NORDIC format
SEIGMT: Converts from NORDIC file to input for GMT
SELMAP: Selects out a part of a MAP file, also creates file for SeismicityViewer
STASEI: USGS station file or ISC station file to SEISAN
USGSNOR: USGS/NEIC CDROM catalog conversion to NORDIC format

**CAT_AGA, reordering of CAT file header lines**
When plotting hypocenters or doing seismic hazard work, it is the first header line in an S-file or CAT-file that is used since it is assumed that it is the prime estimate. When making compact files it is also the first header line, which is used. However, there can be a need for resorting the many type 1 header lines for one or several events so that they are ordered according to agency. It could e.g. be needed to put priority on all the ISC solutions, which then should be the first line in the file. CAT_AGA will reorder the type 1 lines in a CAT file according to the order in which the agencies (3 character codes) are given by the user. If there are many agencies, they can be given in an input file named cat_aga.par, format is one agency per line in the first 3 columns. If the file is not present, the program will ask the user to enter the agencies manually. Optionally, also hypocenter type M lines can be used as type 1 lines. The output file cat_aga.out will contain the sorted events.

**EDRNOR: USGS monthly bulletins (EDR files) to Nordic format**
Program to convert USGS weekly EDR files (ftp://hazards.cr.usgs.gov/weekly/mchedr*) to Nordic format. The program is written by Mohammad Raeesi (email Mohammad.Raeesi@student.uib.no).

**ISCCSV2NOR: ISC CSV catalog to Nordic format**
Program converts csv ascii format to nordic format, csv is used by ISC for catalog data. Only the first 3 magnitudes are used, if a magnitude type in input is b1, output type is x.

**GIINOR, Geophysical Institute of Israel to SEISAN**
The input files are the bulletin type files.

**HARNOR, Harvard to Nordic**

The standard moment tensor solutions given by Harvard (http://www.globalcmt.org/CMTsearch.html) are converted to Nordic format. Strike, dip, rake and moment tensor solution is written out.

The programs can use 3 different input formats:

- Standard format: Default format on screen
- Full format: Full format on screen
- ndk format: File format for downloaded file

The screen formats have the disadvantage that only a limited number of events can be downloaded and captured in one screen so if a lot of events are required, the best alternative is to download a file with all events in ndk format, and then, after conversion, select the desired events.

The ndk format and the full format both give both the hypocenter calculated with arrival times (main header line) and the centroid hypocenter and origin time (MT line), while the standard format does not give the hypocenter calculated with arrival times, so the first header line gives the centroid location. For that reason, it is recommended to use the full format if a screen format is used.

The ndk format does not give Mw which is then calculated from the moment. All formats have additional information not carried over to SEISAN.

**HINOR, Hypoinvers archive format to Nordic**

The input files are the archive type. For details, see HYPINV program.

**HYPNOR, converting HYPO71 files to Nordic files**

Input is just filename of HYPO71 file. A similar program for HYPOINVERSE files is HINNOR.

**HINNOR, converts from Hypoinverse to NORDIC format**

This program works like HYPNOR.

**HSUMNOR, HYPO71 summary file format to NORDIC format**

Note that the program only converts to header lines.

**ISCNOR, converting ISC bulletin file to Nordic format**

This program works with the ISC fixed 96-column format as e.g. distributed on CDROM (files of type FFB). The program can select out subsets of ISC data using a latitude-longitude window, depth and prime magnitude. Any of the magnitudes Ms and mb are used. Before 1978, there was only mb on the CD’s. More detailed selection can be done on the output file later with SELECT. Since the amount of data is very large it is also possible to write out only the hypocenters. Note that ISC now writes in ISF format also, which can be converted with ISFNOR or IMSNOR.

Newer CD’s have compressed data and cannot be used directly. files must be copied to disk first, decompressed and then handled as single files.

The program will first check if a file with agency codes called agency.isc is present. If so the station codes are read from this file (same format as files on CDROM). The program will also check the beginning of the data input file for a possible list of agencies and station coordinates. If present, the stations coordinates are read and converted to SEISAN format and additional codes read in. The agency codes are needed in order to identify in plain text the various agencies used.

Principles in conversion:
Phases: The phases out can be either the phase ID's sent to ISC or the ISC reinterpreted phases (given with a number code in the input file). If the user supplied phases are used, parentheses are removed, and if P/PKP etc is given, it is replaced by P.

Times: If day is incremented relative to origin time day, it is carried into the hours, which can be more than 24.

Agency: It is assumed that it is the same agency for hypocenter and first magnitude. Magnitude is checked for agency; if blank, it is assumed also to be the same as for hypocenter. Only first 3 characters of code is used.

Stations: Only first 4 characters of code are used.

Depth: If no error on depth, a depth fix flag is set.

First motion: Only C or D are used, ISC codes J and B are ignored.

Hypocenter orders: ISC put the best solution last, here the order is reversed, and the prime estimate is first.

Duration magnitude: Change D to C for type.

Distance indicator: If station furthest away is less than 1000 km indicator is L, between 1000 and 3000 km indicator is R and if more than 3000 km indicator is D. If no stations are present the type is set to D.

In order to relocate an event and compare to ISC location, the ISC reidentified phases must be used (option 2, see below). This has the disadvantage that phases not used by ISC (mainly S-phases of local earthquakes) are weighted out in the output file. If option 3 is used, the ISC identified phases are selected if there and if no ISC identification is given, the local reported phase is used. The output file for option 2 and 3 looks the same except that for option 2, the user-defined phases are weighted out. The residuals given in the output file are always relative to the ISC identified phases.

Running ISCNOR:

Below is an example of a run where a latitude - longitude window has been used.

Phases selected can be:
- User reported phases (default=return) : 1
- ISC identified phases only : 2
- ISC identified phases and user reported phases when not identified by ISC : 3

Output: All hypocenters and phases : Return
- All hypocenters : 1
- Prime hypocenter : 2

Latitude range, return for all
60.2 70.5

Longitude range, return for all
10, 20

Depth range, return for all

Magnitude range, return for all
Write selected events on screen (y/n=return)

No agency. isc file present
If ISC CDROM, give drive letter, else return

Give first year and month, e.g. 199501 198601
Give last year and month, e.g. 199602 198602

Now opening d:\1986\198601.FFB
Number of agencies in input file 244

e tc, for each month

  493 events converted
Output file name is iscnor.out
File with stations is isc.sta

The file input can be from a CDROM as in the example above. In that case, the whole CDROM can be read or a smaller time interval can be given. The input can also be from a single file and the program will then ask for the next file when the first has been converted. If many files are to be converted, a list of file names can be made with DIRF and filenr.lis entered as an input file name. The Nordic format output file is iscnor.out and the station list is in isc.sta which has the format used by SEISAN. Optionally, output can also be in the original isc format, however that requires setting a flag in the program and recompiling, see program source code.

**ISCSTA, selecting stations in the complete ISC station file**
The complete station list in the ISC list is very large and it is often an advantage to use a smaller subset, although HYP can use the whole list. The program can select out subsets of stations in both SEISAN and the latest ISC format as found on the ISC home page. The program will read an S-file, find how many different stations there are and select those stations out of a station file, which can either be in SEISAN format or ISC format (automatically determined). The output is in SEISAN format. If no S-file is given the input station file is assumed to be in ISC format and the whole file will be converted to SEISAN format. Optionally, the output can be in high accuracy format, but that is generally only needed for very small networks or small arrays. The latest version of the program supports 5 letter station codes. As of April, 2020, the ISC list contains more then 27000 stations. Hypocenter can work with the 30000 stations.

**NOTE:** The ISC file must only have the stations to remove any text before or after the list.

Start of station list:

<table>
<thead>
<tr>
<th>Code</th>
<th>Station</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Elevation</th>
</tr>
</thead>
<tbody>
<tr>
<td>034A</td>
<td>Hebbronville</td>
<td>27.06470</td>
<td>-98.68330</td>
<td>155.0</td>
</tr>
<tr>
<td></td>
<td>Texas, U.S.A.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>035A</td>
<td>Encino</td>
<td>26.93790</td>
<td>-98.10230</td>
<td>29.0</td>
</tr>
<tr>
<td></td>
<td>Texas, U.S.A.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
KINNOR, Kinematics to NORDIC
Converts .PCK file output of EDPPICK to file in SEISAN format. Many events are converted from one file. The program is based on program from Kinematics by Christopher S. Lim. For info on how conversion is made, see program source code.

ISFNOR, ISF1.0 to and from Nordic
The ISF format is used by the ISC and is an extension to the IMS format. The program is based on the routines provided by the ISC for reading and writing ISF, and the SEISAN standard routines for reading and writing Nordic data. The program converts in both directions. All possible information is converted. Information on the ISF format can be found on the ISC website ([http://www.isc.ac.uk](http://www.isc.ac.uk)). It is recommended to use ISF format for data exchange with ISC.

There is also the newer program IMSNOR which converts from IMS, GSE and ISF to Nordic format. It has been adapted a bit closer to how seis works than ISFNOR.

NORIMS, IMS1.0 to NORDIC format.
The IMS1.0 (International Monitoring System) is a new version of the GSE format and very similar. The program can partly be used for the new ISF (IASPEI Seismic Format) which will include all of the IMS format an additional information needed by ISC and NEIC. The program and the following description is by Mario Villagran. The program works with the IMS1.0:SHORT format (phase-readings/origin files) and the program works both ways.

<table>
<thead>
<tr>
<th>Station</th>
<th>Location</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>035Z</td>
<td>Hargill, Texas, U.S.A.</td>
<td>36.46300</td>
<td>-98.06831</td>
<td>19.0</td>
</tr>
<tr>
<td>058A</td>
<td>Arcadia, Florida, U.S.A.</td>
<td>27.05690</td>
<td>-81.80490</td>
<td>15.0</td>
</tr>
<tr>
<td>059Z</td>
<td>Ave Maria, Florida, U.S.A.</td>
<td>26.33730</td>
<td>-81.44320</td>
<td>8.0</td>
</tr>
<tr>
<td>060A</td>
<td>Indiantown, Florida, U.S.A.</td>
<td>27.03610</td>
<td>-80.36180</td>
<td>9.0</td>
</tr>
</tbody>
</table>

**IMS1.0:SHORT ⇒ Nordic**
Nordic ⇒ IMS1.0:SHORT

The IMS1.0:SHORT format is exactly the one used at the IDC International Data Center (Vienna, Austria). In addition some features used by the ISC International Data Center and the Spanish NDC National Data Center had been added. Magnitudes in IMS format use many characters, the Nordic format allows only one; the following rule is followed:

IMS    Nordic
For mb → 'b'
For MS → 'S'
For ML → 'L'
For MD → 'C'
For MI → 'I'
For MN → 'N'
For mblg → 'G'
For ms → 's'
For MB → 'B'

IMSNOR do not include coda magnitude.

The maximum likelihood magnitudes mb1, mb1mx, ms1, ms1mx, etc are pending. IDC still does not have documentation and they may be changed. Single measurements of magnitude/station are parsed as comment lines (type 3) starting with symbol “$”. When importing data from IMS format, only the “Event IDC” number is parsed and included into a comment line (type 3) of Nordic, together with the ellipse dimensions orientation and the mb standard deviation.

All parameter values read that exceed field limits of Nordic (Amplitude, velocity, snr, etc) have been set to the maximum or minimum possible, example: if snr > 999.9 then snr=999. For conversion from Nordic to IMS it is necessary to use both the hyp.out and print.out files; The reason is that IMS includes many parameters that need to be searched in both files. When converting to IMS format, the user can specify the start numbering for the first event and phase in the file; ignoring will assume (1,1). It is optionally also possible to set the no location flag in the output header lines.

Comments in the IMS input phase data is ignored.

It seems that the program works well for converting ISC data. Just capture the output on the ISC screen and use that as input to IMSNOR. No cleaning of the file is needed before conversion.

**IMSNOR, IMS, GSE and ISF to Nordic format**

Program for reading GSE, IMS and ISF bulletin event files and converting to Nordic format. The intention is that it will replace NORIMS which is doing an incomplete conversion. The program also has several adaptions for SEISAN.

The data from the input file is modified to make it possible to use the data in SEISAN. The source of this data format is mainly CTBTO and ISC. CTBTO is using only the defined IMS or GSE format data while ISC has extended the format to cover other types of data using the ISF format. The program can read both and it is particularly useful for ISC data.

**Assumptions and limitations**

All events are assumed regional

Input files have no component. ISF has an option for component, but it seems it is not used. MULPLT has been modified to plot phases without component.

All error parameters are not read, to be fixed.

**Amplitudes**

Amplitudes are a bit of a mess since very many amplitudes do not use IASPEI standard names or standard methods for measuring and they are therefore useless for further use. The following describe how amplitudes are treated:

All amplitudes which cannot be identified are labeled A. If a time is associated with the amplitude, that time is used. If amplitude is associated with a phase, the time from the phase is used so it is possible to see where the amplitude came from.

IASP amplitude phases are used unchanged, seem to be standard.

LR is a Raleigh wave, amplitude associated with LR and is assumed to be Ms_20.
LQ amplitudes are not associated with a magnitude scale and labeled A.

MLR is non standard, seems to be the same as LR and treated the same way.

Amplitude on a phase line is made into a new line. The time is the same so it is possible to see which phase it was associated with. If possible, it is given and IASPEI standard name (from indication at end of line), if not it is named A.

Ml amplitudes from CTBTO are calculated in a non standard way and cannot be used, give very wrong Ml, labeled A. This might also be the case from other array stations like NORSAR, Norway. No check for other array stations.

Many amplitudes, even on Sg or Lg, are not standard and not labeled IAML. Seems some could be mm WA. If amplitude on P and distance larger than 20 deg, assume IAmb. Some amplitudes have no phase and no time, no time also in S-file.

Amplitudes are mm WA for Nanometrics in Albania system so converted to nm.

Back azimuth

Back Azimuth (BAZ) is put on separate line in order to be able to put in weight. This is needed for ISC where many BAZ values are wrong. ISC does not use BAZ so the ISC observations are identified with no BAZ residual. This has the unfortunate effect that good BAZ are not used, like for NORSAR array or locally determined by 3 components sensors. IMS BAZ is not weighted out since there is a residual. The BAZ is given the name of the phase it was associated with like BAZ-PKP (limited to 8 character length).

Phases

Phases not used by SEISAN are given weight zero, like LR and LQ. More must be added to the list (hardwired).

Phases not used by ISC, as indicated by no residual, is given weight zero. The exception is when ISC does not process the event and use the agency supplied locations. Then there are no residuals. This is identified by 'Event not reviewed by the ISC' which is also given in S-files as a comment. In these cases strange phases might not be weighted out.

Hypocenters

Hypocenters are ordered according to a hardwired table in program. The same order is used for the magnitudes. However, the prime solution (ISC or the last in the list) always come first since residuals are referred to that solution. The priority is: ISC, NEI, GCM, IDC. The list is hardwired in routines sort_source_info.

Relocating ISC events gives higher rms with SEISAN. Part of the problem is that epicentral distance is different, at certain distances up to 100 km. The travel time model is also different. Using residual weighting gives similar rms to ISC, might be what ISC is using.

Fault plane solutions

Fault plane solutions are also ordered, different table. The priority is: GCM, NEI and ISC.

Magnitudes

Only standard magnitudes are put on header lines, same order as for hypocenters. The first 5 magnitudes are put on first hypcenter, the rest distributed among following hypocenters and if more put in as comments. The first magnitude is repeated in 3. position on header line, therefore only 5 different. Non standard magnitudes are listed in comments.
Macroseismic data is not included, but if event is felt, it is marked with a blank type 2 line.

Event type

IMS use the following characters to define event types:

The leading characters are
s = suspected
k = known
f = felt (implies known)
d = damaging (implies felt and known)

The trailing characters are
c = meteoritic event
m = mining explosion
e = earthquake
n = nuclear explosion
h = chemical explosion
r = rock burst
i = induced event
x = experimental explosion
l = landslide

In SEISAN they are converted to event types using the 2 or one letter(s):
se to E = mining explosion
n to E = nuclear explosion
sh to P = probable explosion
ke to Q = known earthquake
se to ‘’ = probably earthquake
sl or kl to L = probably or known landslide
si or ki to I = probably or known induced event
f or d indicated as a felt event (type 2 line)

All remaining IMS types are ignored (blank)

NORCSV, NORDIC to CSV format

The program converts the date and time, hypocenter location and magnitude to a simple CSV file that can be loaded into spreadsheet software. An example of a line is

2012/01/06,06:16:26.80, 76.191, 9.870, 10.0, 2.3,CBER

where the columns are date, time, latitude, longitude, depth, magnitude and magnitude type. The magnitude with type of the highest priority is selected as given by the magnitude order in SEISAN.DEF.
For example ML from your agency may be given priority over MC, or you may prefer your ML over that from another agency.

**NORGSE, NORDIC from and to GSE parametric format**
The program (written by Mario Villagrán) converts parametric data between Nordic and GSE2 format. It can be used interactively or by giving the options as arguments. Type `norgse -help` to see the options.

**NORHIN, From Nordic to Hypoinverse archive format**
The program is started by typing `norhin input-file`. The output file is `norhin.out`. For more details, see program `HYPINV`, `indexHYPINV`.

**NOR2QML** From Nordic to and from Qml. The program has many options, see below. To simplify, there are two scripts:

- NOR2QML `filename`
- QML2NOR `filename`

They have some fixed options so if only converting one file in working directory, they are the easiest to use.

**NORHYP, From Nordic to HYPO71 format (SUN and PC)**
The program is written by F. Courboulex. The program asks for the input file name and the output file name is `norhyp.out`.

NOTE, the program expects there to be a type 7 line for each event, else it will not output anything.

**PDENOR, converting PDE bulletin file to NORDIC format**
PDE distributes bulletins on e-mail, both a monthly bulletin and a weekly bulletin (different formats). The program converts one of these files to Nordic format and put the file into a standard SEISAN database called PDE for the monthly files and PDEWE for the weekly files. This database must have been created before running the program. Both CAT and S-files are made and SELECT and EEV can be used afterwards. Files can be received by email or picked up at hazards.cr.usgs.gov, dir pde for monthly files and weekly for weekly files, must be ehdf type.

**RSANOR**
Program converts between format used by “Red Sismologica de Andalucia” and a few others in Spain.

**NOR2QML**
NOR2QML is an online service that enables conversion of data in NORDIC format into QuakeML and QuakeML to NORDIC format.

The service is found at [http://nnsn.geo.uib.no/nor2qml/](http://nnsn.geo.uib.no/nor2qml/)

**Nordic → QuakeML**
Conversion from Nordic format to QuakeML is found at [http://fnsn.geo.uib.no/seis2qml/sfile-to-quakeml](http://fnsn.geo.uib.no/seis2qml/sfile-to-quakeml).
Here one needs to select a file in Nordic format on the local harddisk, next select "UPLOAD ALL". A new window will open where one can choose QuakeML version, prefix and authority ID. Authority ID is important if the data will be made public. Next choose "GENERATE XML".
Type in a filename for the QuakeML file that is generated and select "SAVE FILE".

**QuakeML → Nordic**
Conversion from QuakeML to NORDIC format is found at [http://nnsn.geo.uib.no/seis2qml/quakeml-to-sfile/](http://nnsn.geo.uib.no/seis2qml/quakeml-to-sfile/)
There are three options:
1. "Web-Service link” provide conversion from third party QuakeML web services. Type in the URL
from a web service in the box (see "Click here for an example") and select "GENERATE".
2: "XML text" converts plain text QuakeML into Nordic format. Paste the QuakeML format into the
box and select "GENERATE". In the new window enter name of output file and select "SAVE FILE",
below is the Nordic format shown.
3: "XML file" converts QuakeML files the Nordic format, select a QuakeML file on the local harddisk
and select "UPLOAD". In the new window enter name of output file and select "SAVE FILE", below is
the Nordic format shown.

Please report bugs to the developer, Christian Rønnevik (Christian.Ronnevik@uib.no)

SEIGMT, Nordic to GMT input
The program SEIGMT reads information from Nordic or compact files and writes the parametric data
to files that can be used as input for GMT (Generic Mapping Tools, [http://gmt.soest.hawaii.edu/]).
The user can choose a scaling for the magnitudes and also select a magnitude type order. The scaling
option is useful if you wish to scale the symbol size of your epicenters with magnitude. The magnitude
type order defines, which magnitude should be taken in case several magnitudes have been determined
for one event. If you don’t give a magnitude order, the program chooses the largest magnitude.

The files written by SEIGMT are:
gmtxy.out - event locations, to be plotted with psxy
gmtxyz.out - event locations and depths, to be plotted with psxy
gmtstxy.out - station coordinates (longitude, latitude and station code)
gmtpath.out - travel path data, to be plotted with psxy
psmeca.out - fault plane solutions, to be plotted with psmeca (Aki and Richards convention)

SELMAP, selecting a subsection of a MAP file
The program can retrieve parts of a large MAP file written in SEISAN map format. On the SEISAN web
site or on the SEISAN CDROM, very detailed global mapfiles are available in SEISAN format. The file
originally comes from the USGS. SELMAP can select out part of a MAP file in a latitude-longitude grid.
The MAP files consist of several small segments and a segment is selected if at least one point is inside the
specified grid. The program also creates an output file in xyz format for the Lomax SeismicityViewer. The
program can output a smaller number of points than available in the input file by using the parameter
skip.

STASEI
Converts the official global station file from USGS (comma format) or ISC global station file to SEISAN
station format (same as HYPO71 format with SEISAN extension for 5 letter station codes). A list of
most global stations are now found on the SEISAN CD. It seems that the USGS format is no longer used.

USGSNOR, USGS catalog to NORDIC format
The program converts USGS CDROM hypocenters to NORDIC format. Most of the information is used.
If more than 3 magnitudes are available, only the 3 first are used. The number of stations is included
when available. The depth is indicated as fixed in all cases where the operator has been used (A,N,G).
Macroseismic information is included with max intensity. The residual standard deviation is put into rms
column. Event types are set to R. Magnitude types are converted as follows:
UK is made blank
b is replaced by B
s is replaced by S
D is replaced by C
w is replaced by W
CHAPTER 19. FILE CONVERSION AND MODIFICATION PROGRAMS

WAVEFORM CONVERSION PROGRAMS

This group of programs are mostly converting waveform files from some format to SEISAN although a few also convert from SEISAN to some other, mostly standard, formats. Most programs convert from binary to binary formats.

Many instruments come with conversion programs to some standard format like PCSUDS or MINISEED, and these have often been used to convert to SEISAN instead of writing programs reading the original files directly. Many such conversion programs work on PC so the corresponding SEISAN programs only work on PC. However, since the PC files can be read directly on Sun, this should not present a problem. Many programs have VERY LITTLE documentation, look in source codes for more information.

The number of programs are forever increasing with new recorders coming onto the market and new formats coming in use and others going out of use and it is becoming increasingly difficult to keep track of it all. For this release of SEISAN it has not been possible to test all programs on all platforms. An attempt has been made to standardize the programs. A general problem is that many seismic recorders and formats do not provide proper identification of the channels. In the worst cases, there are no station codes, only channel numbers and in very many cases, there is no room for proper component information. This is being taken care of by having a definition file, and only one format for the definition file is used, see below. This is also used with program WAVFIX.

Most programs work in the standard way with a \texttt{filenr.lis} file as input (made with DIRF).

The response information is seldom in the original files and in most conversion programs, the response information is taken from the CAL directory. If no response information is available, a message will be given. For each program, a comment will be given as to the status of testing and on which platforms they operate. If the channel definition file option is implemented, the array dimensions will be SEISAN standard.

The program SEIPITSA might be an easy way to convert between 1-column ASCII data and SEISAN (see below).

When converting between the major analysis format (MiniSEED, SEISAN, SAC and GSE) mostly using program WAVETOOL, only SEISAN and MiniSEED will preserve the network and location codes as well as the flag for uncertain timing since the other formats only partly have room for this information.

Conversion programs definition file

The conversion programs use a common format for the definition file for naming station and channels. The definition file is named programname.def as e.g. sudsei.def. The definition file can be in the working directory or the DAT directory. The conversion program will first look in the working directory for the file and then in DAT. The conversion of codes can take place in 2 ways (see below for details): (1) An input station and component code is converted to an output station code and component. (2) an input channel number is assigned a station and component code. The advantage of (1) is that the conversion is independent of the channel number or order, however, the user must then know the default station and component names generated by the conversion program.

Default assignment of station code and component:

This is very much dependent on the conversion program used since some data files have complete information and others very little, see description of individual programs in manual or at start of source codes. In all cases, the conversion program will make both station and component codes based on what is available of information in the input files. IT IS THESE CODES THAT are used for input code as described below. In order to find out what they are, it is easiest to run the conversion program once (without a def file) and see what codes the program assign. Alternatively, some of the programs have documentation in the manual. Some of the station codes might be instrument serial numbers, which are not always known. Therefore, running a test might be the best way to find out.
In addition to converting channel codes, the def file can also give SEISAN waveform file header information and network code as it appears in the file name. If no network code is given, the network code will be the station code of the first channel.

Principle of conversion in order of precedence:

1. Both station and component given on input: Converted to what is given for output station and component.

2. If both are not present, the channel number is used.

Header line text (29 char)... NetCd (5 chars), Comment for next line
Header for REFTEK NEWNT
chan stati comi stato como, In and output definitions, comment for next line

<table>
<thead>
<tr>
<th>chan</th>
<th>stati</th>
<th>comi</th>
<th>como</th>
<th>stato</th>
<th>como</th>
</tr>
</thead>
<tbody>
<tr>
<td>BO11</td>
<td>S</td>
<td>Z</td>
<td>B</td>
<td>Z</td>
<td></td>
</tr>
<tr>
<td>BO12</td>
<td>S</td>
<td>N</td>
<td>B</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>BO13</td>
<td>S</td>
<td>E</td>
<td>B</td>
<td>E</td>
<td></td>
</tr>
</tbody>
</table>

The first line is just a comment line, must be there in any format. Here it shows where the network code is positioned as indicated by NetCd.

The second line gives the header information for the SEISAN main header, which are the first 29 characters. The file name network code is also given and is here NEWNT. Format a29,1x,a5.

The third line is just comment to indicate the position of the columns in the following lines (max 200). A line must be there. The abbreviations are:

- chan: Channel number, optional unless no input station and component given.
- stati: Input station code, 1-5 chars
- comi: Input component code, 4 characters
- stato: Output station code, 1-5 characters
- como: Output component code, 4 characters. First character MUST be S, L, B, A, or I, last character MUST be Z, N or E, all upper case.

Format i5,1x,a5,2x,a4,1x,a5,2x,a4

The conversion programs are listed below
AFADSEI: AFAD acceleration records to SEISAN
AHSEI: AH ASCII to Seisan, little tested
ARCEI: Reftek archive to Seisan, windows only
ASCSEI: ASCII to SEISAN
BGISEI: Beijing Geodevice Institue to SEISAN
CITSEI: CityShark to SEISAN
CNVSSA: Kinemetrics SSA to Kinemetrics Dataseis
CNVSSR: Kinemetrics SSR to Kinemetrics Dataseis
DELS: Deletes specific parts of an S-file
DIMASSEI: USGS Dimas to Seisan
DRSEI: Sprengnether recorders to SEISAN
GIISEI: Geophysical Institute of Israel to SEISAN
GSRSEI: GeoSig to SEISAN
GSESEI: See WAVETOOL
GSERESP: Conversion between GSE and SEISAN response files
GURSEI: Guralp to SEISAN
GCF2MSD_MANY: Run Guralp gcf2msd many times from files in a file structure
IRISEI: From IRIS ASCII to SEISAN
ISMEI: ISMES to SEISAN
KACSEI: Kinemetrics ASCII acceleration to SEISAN
KINSEI: Kinemetrics Dataseis to SEISAN
K2SEI: Kinemetrics K2 to SEISAN
LEESEI: Willy Lee system to SEISAN
LENPCQ: Converts from Lennartz to PCEQ to PCEQ format
LENSEI: Lennarts ASCII to SEISAN
M88SEI: Lennartz MARS88 to SEISAN
NANSEI: Converts from Nanometrics to SEISAN
NEISEI: Converts from NEIC CDROM waveform data to SEISAN
NRWSEI: Geol. Survey. of Northrhine-Westphalia format to SEISAN format
OS9SEI: Converts SEISLOG files to SEISAN waveform files
PITSA: Conversion programs described with program PITSA
PCQSEI: Converts from PCEQ to SEISAN
PDASEI: Geotech Instruments PDAS to SEISAN
PSNSEI: Public Seismic Networks to SEISAN
QNXSEI: SEISLOG QNX to SEISAN
RDSEED: IRIS program to read SEED volumes
RSASEI: Conversion from Andalucian Seismic network to SEISAN
RT_SEIS: Reftek Passcal format to SEISAN conversion
SEI2PSXY: Convert waveform data to trace input for psxy
SEIM88A: Conversion from SEISAN to MARS88 ASCII format
SEISAN2MSEED: From SEISAN to MiniSEED
SEISAF: SEISAN to SESAME ASCII
SEIPITSA: SEISAN ⇔ PITSA ASCII
SGRSEI: SeisGram to SEISAN
SISSEI: Sismalp format to SEISAN format
SILSEI: SIL network ASCII files to SEISAN
SUDSEI: PCSUDS to SEISAN
TERSEI: Terra ASCII to SEISAN
VELNOR: Velest CNV to Nordic
WGSSEI: WGSN format to SEISAN
WAVASC: Waveform file to 2 column ASCI file
For each program, a summary of capabilities is mentioned: The platforms available (all for all or specific name), channel definition file available (chan. def. yes or no) and if the program will look for response files in the CAL directory to insert in the headers (resp. yes or no).
If you do not find the conversion program here, look on the ORFEUS website for other programs that might convert to one of the formats used above. ([http://orfeus.knmi.nl/other.services/conversion.shtml](http://orfeus.knmi.nl/other.services/conversion.shtml)).

**AFADSEI, AFAD acceleration to SEISAN or MiniSeed**
- all, chan. def. yes, resp yes
Converts from AFAD ASCII acceleration to SEISAN or MiniSeed. Components are hardwired to HN Z etc. Assume 3 channels files only, all channels same sample rate and number of samples. Input values are in gal, the output is in 1/10 000 gal or 0.001 mm/s², station code is taken from file (a number). There is an output file with hypocenter info from input files as well as the waveform file name. For each channel a response file is written in SEISAN format.

**AHSEI, AH ASCII to SEISAN**
- all, chan. def. yes, resp yes
Converts AH ASCII files to Seisan format.

**ARCSEI, Reftek archive to SEISAN (windows only)**
ARCSEI is a program to automate the extraction of data from a RefTek data archive and the conversion to SEISAN format. The program works interactively and with a simple text interface that asks the user to enter the details for the data extract. Based on the user selected criteria the program (1) extracts the data from the archive in Passcal format using ARCFETCH, (2) converts the Passcal data files to SEISAN format using RT_SEIS, and (3) merges the SEISAN files if merging is activated by the user, using SEISEI. The program is written in Fortran and works on Windows only.

The ARCSEI program can be used in various ways:

- to extract a single time window from one or several stations
- to extract several time windows from one or several stations
- to extract sequential time windows from one or several stations

The ARCFETCH and RT_SEIS programs, both part of the RefTek software package, have to be installed (see RefTek documentation) and the PATH variable set to include the directory where the programs are stored. It is assumed that the RefTek data archive exists and that the user is familiar with the content of the archive. The archive content can be shown with the command ARCINFO. To test that the program is installed correctly, open the Windows command tool (from the menu, or by selecting Start → Run → cmd) and type ARCSEI <RETURN>.

The definition file: arcsei.def

The purpose of the definition file is to set some parameters needed to run ARCSEI, however, the program also works without. The arcsei.def file can either be stored in the seismo/DAT directory, or the current working directory. The program first checks in the current directory. The arcsei.def file should be adjusted to the user’s set-up, before ARCSEI is started.

The parameters are:
**ARCHIVE:** The path of the RefTek data archive, can also be entered manually at run time.

**OUTPATH:** The directory in which the SEISAN files are to be stored. The default is `\` (the current directory).

**MERGE:** Select if SEISAN files from several stations for the same time interval should be merged (Y), or not (N).

**NETWORK_CODE:** Network code used in case SEISAN files are merged.

**CHANNEL:** Data channel in RefTek archive consisting of the unit, stream and channel (unit,stream,channel). The * can be used as wildcard to select all streams or channels, BUT not to select all units (since ARCFETCH is used in cooked mode, which means that the time interval extracted matches the input start- and end-time).

---

Example of the arcsei.def file

<table>
<thead>
<tr>
<th>KEYWORD</th>
<th>Comments</th>
<th>Par1</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCHIVE</td>
<td>Path to archive</td>
<td>G:\CTBTO\ARCHIVE</td>
</tr>
<tr>
<td>OUTPATH</td>
<td>Path to put</td>
<td>C:\test</td>
</tr>
<tr>
<td>MERGE</td>
<td>Y or N for merging Seisan files</td>
<td>Y</td>
</tr>
<tr>
<td>NETWORK_CODE</td>
<td>used for merging</td>
<td>ARCSE</td>
</tr>
<tr>
<td>CHANNEL</td>
<td>Name of channel</td>
<td>8020,1,*</td>
</tr>
<tr>
<td>CHANNEL</td>
<td>Name of channel</td>
<td>8021,1,*</td>
</tr>
<tr>
<td>CHANNEL</td>
<td>Name of channel</td>
<td>8022,1,*</td>
</tr>
<tr>
<td>CHANNEL</td>
<td>Name of channel</td>
<td>8023,1,*</td>
</tr>
<tr>
<td>CHANNEL</td>
<td>Name of channel</td>
<td>8024,1,*</td>
</tr>
<tr>
<td>CHANNEL</td>
<td>Name of channel</td>
<td>8025,1,*</td>
</tr>
<tr>
<td>CHANNEL</td>
<td>Name of channel</td>
<td>8028,1,*</td>
</tr>
</tbody>
</table>

---

**How to run the program:**

ARCSEI is started from the Windows Command Tool (cmd) by typing

```
arcsei <RETURN>
```

**ARCSEI - DATA EXTRACTION FROM REFTEK ARCHIVE AND CONVERSION TO SEISAN FORMAT**

SEISAN FILES ARE MOVED TO DIRECTORY: C:\test

ENTER ARCHIVE, OR <RETURN> TO USE DEFAULT (default: G:\CTBTO\ARCHIVE)

Return to accept default, which is set in the arcsei.def file, or give the archive path.

AVAILABLE CHANNELS ARE:

8020,1,*
Type channel and <RETURN>, if defined in arcsei.def channels are listed, otherwise an example is shown. The channel is given as unit,stream,channel. Wildcards can be used for stream and channel, but not for the unit.

**NEXT CHANNEL OR RETURN TO CONTINUE**

Additional channels can be entered, to continue press <RETURN>.


Examples: 2000:200:12
2000:200:12:15
2000:200:12:33:15

Type start time as year:day-of-year:hour:minute:second. Minute and second can be omitted.

**NEXT START-TIME OR RETURN TO CONTINUE**

Additional start times can be entered, to continue press <RETURN>.

**ENTER END TIME USING ONE OF 3 OPTIONS:**
- Absolute time as YYYY:DDD:HH:MM:SS (like start-time)
- +seconds for time interval (e.g. +300)
- ++seconds for multiple intervals (continuous extract, e.g. ++300)

Specify the end time, either in the same style as for the start time (only if one start time), or in some cases more useful, the desired time window in seconds, by entering +seconds. If sequential time windows are to be extracted, use ++ seconds. The user is then asked how many time windows should be extracted. It is thus possible e.g. to extract 10 consecutive windows of 600 seconds. Only if sequential extract windows specified:

**ENTER NUMBER OF CONTINUOUS WINDOWS**

After the program has finished, the data in SEISAN format can be found in either the current directory (default) or in the OUTPATH directory if the variable is specified in arcsei.def. The temporarily created files are deleted automatically.

*How it works*
ARCSEI reads the user input that specifies what should be extracted from the RefTek archive and then calls the programs ARCFETCH, RT_SEIS and SEISEI. For temporary data storage ARCSEI creates the directory arcsei_temp under the current directory. The arcsei_temp directory is automatically deleted upon program completion.

1. Create empty arcsei_temp directory

2. Arcfetch

The arcfetch program performs the data extraction from the RefTek archive. A complete list of the command line input of arcfetch can be obtained by starting the program without additional options. ARCSEI starts arcfetch in the following way:

```
arcfetch archive channel,start-time,end-time -o OUTPATH -c
```

Where:

- `-o OUTPATH`: Specifies the output path for arcfetch, always arcsei_temp
- `-c`: Specifies cooked mode, which means that the time interval extracted matches the input start- and end-time (this is not the case, when not running in cooked mode)

Example:

```
arcfetch G:\ARCHIVE 8020,1,*,2000:200:12,+10 -oarcsei_temp -c
```

3. RT_SEIS

RT_SEIS converts all files with the suffix 'rt' in arcsei_temp to SEISAN format. RT_SEIS reads the RTU.INI file for station definition, if the environmental variable RTU is set to point to the RTU.INI file (see RT_SEIS section below).

4. SEISEI

SEISEI, if merge is selected, merges all SEISAN files in the arcsei_temp directory.

5. move

Finally all files (single or merged) are moved to the OUTPATH directory or the current directory if OUTPATH is not defined. In case multiple stations are selected, ARCSEI runs steps (1) and (2) in a loop, before the data is merged and moved. In case several time windows are selected, ARCSEI runs steps (1) to (4) in a loop, and in addition a second loop over multiple station (1) and (2). If sequential time windows are specified, ARCSEI computes multiple start times and works as if these time windows were user specified. All, def. File yes, resp yes

**ASCSEI, ASCII to SEISAN**

```
all, chan. def. yes, resp yes
```

Converts a single column file to a one channel SEISAN file. Date, time, sample rate (number of samples per second), station and component must be entered manually. The user is ask for a scale factor, normally it is 1.0. If input numbers are smaller than 1.0, a scale factor must be used since numbers are written out as integers. The input file must contain only a column of numbers and no headers.

Example:

```
linux$ head data.txt
-1646
-1722
```
linux$ ascsei
File name, # or filenr.lis for all
data.txt
data.txt
year, month, day, hour, min, sec
2017,02,22,11,30,05.12
sample rate
20
station, max 5 chars
JENS
component, max 4 chars
BH Z
Enter scale factor, return for no scaling

Number of samples in file: 144132
Outfile name: 2017-02-22-1130-05S.JENS__001_BZ
JENS 1117 53 2 22 11 30 5.120 7206.550
No response file for this channel --------
JENS BH Z117 53 2 22 11 30 5.120 20.00 144132

BGISEI, Beijing GEODEVICE FORMAT (BGI) to SEISAN. Linux, PC, chan. def. yes, resp yes
The program to convert waveform files from BGI to SEISAN format is called BGISEI. The instrument
response in the original files is not used. The program has only been tested with data recorded in Cuba.
The program is written by Bladimir Moreno.

CITSEI, CityShark to SEISAN all, chan. def. yes, resp yes
Converts from CityShark ASCII to SEISAN. Components S Z, S N, S E are assumed for the 3 channels.
Assume 3 channels files only, all channels same sample rate and number of samples.

CNVSSA and CNVSSR Kinematics accelerometers to Kinematics Dataset
PC
The programs are supplied by Kinematics to convert from SSA and SSR formats to Kinematics Dataset.
To further convert to SEISAN, use program KINSEI. Only PC executable programs are available. The
data is 16 bit.

CSS
At the moment there is no direct conversion from CSS to SEISAN. It is possible to convert CSS data to
SAC or GSE using other tools like codeco, Geotool and SAC, and then convert to SEISAN format.

DELS, deletes parts of S-file

DELS, program to delete specific lines in an S-file with one or many events
There is often a need to delete particular parts of an event file like the spectral lines or the P-phases etc.
Below is an example run.

C:\seismo\WOR>dels

Give input file

hyp.out

Give line to delete or keep, terminate with enter
1: Lines with P-phase Not delete
2: Lines with S-phase Not delete
3: Lines with SPEC-phase Not delete
4: Lines with IAML-phase Not delete
5: Lines with IASP-AMP phase Not delete
6: Lines with AM, AT and AS phase Not delete
7: Lines with END phase Not delete
8: Lines with BAZ phase Not delete
9: Lines with any phase Not delete
10: Lines with fp solutions Not delete

Any combination of options can be selected or deselected. The output is in file dels.out. If used within EEV, it is used with current event S-file.

A special option is 99. This will write out IAML amplitudes for Z, N and E if all three present. Output file is dels.amp.out. The program can also be used with arguments

First argument is file name
Second argument is optionally 'overwrite' meaning that input file will be overwritten, can only be used with a file with one event, if more program stops. Third argument is operator

The last two arguments are used with EEV, command dels in EEV.

**DIMASSEI, USGS DIMAS to SEISAN** all, chan. def. yes, resp yes
Converts Dimas files to Seisan format.

**DRSEI, Sprengnether data recorders to SEISAN** all, chan. def. yes, resp yes
Converts Sprengnether DR3024 and DR3016 to SEISAN format. These two formats are slightly different, but the program makes the adjustment. Only essential information is read in and only 4 lowest digits of serial number are used. If station codes are set up, these are used, else the serial numbers are used for station codes.

**GIISEI, Geophysical Institute of Israel to SEISAN** all, chan. def. yes, resp yes
Converts Geophysical Institute of Israel imported DAQ files to SEISAN format. The initial station codes are as defined in file, can be converted with the normal .def file. If 4.th character of station name indicates component (N or E), that is blanked out and transferred to 4.th character of component name BEFORE using the def file conversions.

**GURSEI, Güralp to SEISAN** all, chan. def. yes, resp yes
Converts Güralp GCF files to SEISAN format, only works with one channel data. Maximum number of samples as defined in seisan, at least 1 200 000, channels codes can be defined using the gursei.def definition file. If no definition file, the station name is the first 4 letters from internal station name and the component is BH Z.
GCF2MSD_MANY
Runs Güralp gcf2msd many times from files in a file structure. It is only tested for one particular structure with 15 min files, so the program has to be adopted to other structures. Requires program gcf2msd from Güralp. GCF2MSD_MANY

GSERESP, conversion between GSE and SEISAN response files
The program provides conversion between SEISAN, GSE1 and GSE2 response files. The response can be given in frequency, amplitude and phase (FAP) triplets or in poles and zeros (PAZ). Since the number of values in the GSE format is unlimited the conversion from SEISAN to GSE only changes the format, whereas converting from GSE to SEISAN, if the number of FAP triplets is more than 30 or the number of poles and zeros larger than 37, the response in SEISAN format will be approximated by 30 FAP triplets. The output files in SEISAN format will have the default SEISAN response filenames (see RESP program and SEISAN response format). Output files in GSE format will include the station name, the component, number 1 or 2 for GSE1 and GSE2 respectively and end on '.CAL' (e.g. MOR_SHZ2.CAL (GSE2), KONO_BZ1.CAL (GSE1).

GSRSEI, GeoSig to SEISAN
Converts from GBV recorders to SEISAN. GeoSig was earlier GeoSys. Before version 8.1, there was a bug in program so start time was wrong by the amount of the prevent time.

IRISEI, IRIS ASCII to SEISAN
The input format is the variable ASCII download format used on the GSN Quanterra stations. The format is used in connection with SEISNET. The program only works if input file has more than 1000 samples. ISMSEI, ISMES to SEISAN PC, chan. def. no, resp no ISMES is an Italian seismic recorder. This is the first version of the program made by IIEES in Iran. The program can convert one file with up to 3 channels.

KACSEI: Kinemetrics ASCII acceleration to SEISAN
Kinemetrics ASCII film record acceleration files (type *.v1) to SEISAN. It is assumed that:
- channel 1 is N, 2 is Z and 3 is E
- there are always 3 channels in file
- input values are in 1/10 g, the output is in 1/1 000 000 g
- station code is taken from file name as given in first line of input file
- the 3 channels can have different number of samples, however it is assumed that they all start at the same time

KINSEI, Kinemetrics DATASEIS to SEISAN
The program takes the station code from the input files. The component codes are also taken from the input file as far as Z, N and is E is concerned, but the first letter is always set to S, like 'S Z'. The program is also used if CNVSSR or CNVSSA have been used first.

K2SEI, Kinemetrics K2 to SEISAN
Program for K2 binary files. The program works by first converting the binary files to ASCII by internally running the Kinemetrics program kw2asc (PC only). If no definition file is present, channel 1-3 will be A Z, A N and A E. If more channels they will be called A 04, A 05, etc.

LEEESEI, Willy Lee binary files to SEISAN
The number of channels is fixed to 16 and the time information is not read, it must be entered when converting the file.
CHAPTER 19. FILE CONVERSION AND MODIFICATION PROGRAMS

LENSEI, Lennartz ASCII to SEISAN all, chan. def. yes, resp yes

LENPCQ, converting Lennartz to PCEQ format PC
Only executable code for this program and only PC (made by the Royal Belgian Observatory). The format is used by an older version Lennartz tape recorder. The output files have the same names as the input files and are placed in a directory c: qcola, WHICH MUST BE THERE.

M88SEI, Lennartz MARS88 to SEISAN all, chan. def. yes, resp yes

NANSEI, Nanometrics to SEISAN PC, Sun, chan. def. yes, resp yes
The program converts from the Y-file format to SEISAN. This is done by first making an ASCII file with Nanometrics y5dump program (done internally in NANSEI). NOTE: The y5dump program requires some special Nanometrics libraries (Solaris) or *.DLL files (PC), which are included and installed with SEISAN (see installation section). The program converts single channel files only.

NEISEI, NEIC digital data to SEISAN PC, chan. def. no, resp no
NEIC earthquake digital data comes on CDROM. The data is extracted with a program coming with the data and then converted to SEISAN binary waveform data. The response information is given as poles and zeros in the SEISAN waveform file header.

OS9SEI, converting SEISLOG files to SEISAN PC, SUN, chan. def. no, resp yes
The program takes a SEISLOG ASCII (downloaded in CMP6 format) or binary file and converts to a SEISAN file. The input can be several files from a filenr.lis or an ASCII downloaded file either compressed or uncompressed. The program will look for the calibration file in the CAL directory and add it to the SEISAN file, or give a message if it is not there. The program will work with SEISLOG files recorded under operating system OS9 or QNX up to version 7.6. For QNX version 7.0, use program QNXSEI.

PCQSEI, converting PCEQ format to SEISAN PC, chan. def. yes, resp no
PCQ format to SEISAN. Earlier used with IASPEI software libraries.

PDASEI, converting PDAS files to SEISAN all, chan. def. yes, resp yes
The program converts a single channel PDAS file to a single channel file in SEISAN format. Several of these files can then be merged with SEISEI. PDASEI in previous SEISAN versions (before version 6.0) only worked with PDAS in 16-bit format, so if 32 bit or gain ranged format was input, the output would have been in error. The current version of PDASEI should be able to convert all 3 types of input files. A description of the PDAS format is found in the PDASEI program.

PSNSEI, Public Seismic Networks to SEISAN all, chan. def. yes, resp yes
The Public Seismic Network recording system makes one file pr channel. Since component is not well defined, several files from the same recording system might get the same SEISAN file name. Do some testing when setting up the recording system. The one component files can be assembled into multichannel files with SEISEI. There might be a newer version of PSN format not supported.

QNXSEI, SEISLOG QNX version to SEISAN all, chan. def. no, resp yes
This program works as OS9SEI except that it does not read the ASCII files. The program must be used with Seislog 8.0. The program is currently the only program that put in the time synchronization flag in SEISAN waveform files except for data logging programs under Seislog Windows. See format description in Appendix B. The program recalculates the sample rate based on the time in the first blocks in the file and the last blocks in the file (each block is one second long). For very long files, this might be of importance since the digitizer might not have exactly the nominal sample rate.
RSASEI, Andalucian Seismic Network to SEISAN  
All, chan. def. yes, resp yes
Conversion of network and broadband files to SEISAN format. Covers several versions of the DTS format also used by other institutions in Spain. Not tested on Linux.

RT_SEIS, Reftek Passcal to SEISAN  
PC, chan. def. no, resp no
The RT SEIS program converts Reftek Passcal format to SEISAN. This program is provided by Refraction Technology Inc. The program does not use the filenr.lis as input file. To see the options of RT_SEIS, start the program without any arguments. In order to make use of the RTU.INI definition file, the environmental variable RTU needs to be set to for example c:
seismo
dat (see RefTek documentation for more details). This file can be used to set station names for respective unit IDs.

Example of RTU.INI:

[8020]
Station=SB00
Network=CTBTO
CH1Band=
CH1Type=
CH1Axis=a
CH1Loc =
CH2Band=
CH2Type=
CH2Axis=b
CH2Loc=
CH3Band=
CH3Type=
CH3Axis=c
CH3Loc =

[8021]
Station=SB01
Network=CTBTO
CH1Band=

...

SEI2PSXY
Converts waveform file to GMT psxy trace plotting ASCII file. The output files have one line for each sample giving the date and time and amplitude value, e.g.:
2005/06/16T00:59:59.51 -40.0000
Each trace will have its own file and the output names are given for each like
Output file name: 2005-07-31-1544-10S.STOK_.003_STOK_S_Z
Output file name: 2005-07-31-1544-10S.STOK_.003_STOK_S_N
Output file name: 2005-07-31-1544-10S.STOK_.003_STOK_S_E
To plot the trace data with psxy, use projection ‘-JX<xsize>T<ysize>’ and option ‘-R’ giving time range in the same style as the data. To plot the data the gmtdefaults should be set to ‘gmtset INPUT_DATE_FORMAT yyyy/mm/dd INPUT_CLOCK_FORMAT hh:mm:ss.xx’. See psxy man pages for more details.
SGRSEI  PC, chan. def. yes, resp yes
SeisGram binary to SEISAN. Only 3 component data has been tested. Channel order is assumed to be Z, N, E. The input real values have been multiplied by 100 000 before being converted to integers. Program little tested.

SEED
The Standard for Exchange of Earthquake Data (SEED) format is defined by the Federation of Digital Seismographic Networks (FDSN). The rdseed program is distributed with SEISAN to extract data from SEED volumes. RDSEED is an IRIS program to read SEED volumes. The program provides conversions to SAC (ASCII and binary), AH, CSS and miniseed. It is described in the file ‘rdseed.txt’ in the INF directory. Updated versions of rdseed will be available at http://www.iris.washington.edu/pub/programs. A PC version (rdseed.exe) is distributed with SEISAN CD (also on home page). SEED volumes contain the complete response information, details on how to convert the SEED response to GSE response format can be found in Havskov and Alguacil [2004].

SEIM88A, conversion from SEISAN to MARS88 ASCII format  all, chan. def. no, resp no
The program converts SEISAN waveform files to Lennartz-ASCII MARS88 format. The program will write one file per channel. Output files are either mars.xxx if a single file is converted or marsxxx.yyy if the ‘filenr.lis’ file is used as input.

SEIPITSA  all, chan. def. yes, resp yes
The program converts from SEISAN to PITSA ASCII format and back. The ASCII format has one file per channel. The user will be asked for a name of the output file-system. If a single file is converted, the channel number will be added to the output file-system name (e.g. data.001). If the ‘filenr.lis’ file is used the filenumber will be added to the file-system name (e.g. pitsa001.004, first file and fourth channel). The program is no longer used for conversion when PITSA is started from EEV, but might be useful, since it creates one column ASCII data and can easily be modified.

SEISAF, SEISAN to SESAME ASCII  all, chan. def. no, resp no
The 3 first channels in SEISAN file are read. There is no check if from same station. It is assumed that the order in SEISAN file is Z,N,E, that all 3 channels have the same start time, number of samples and sample rate. These values are taken from the first trace.

SEISAN2MSEED  All chan.def. no resp no
By Chad Trabant, IRIS Data Management Center
Program developed at IRIS to convert from SEISAN to mseed, all platforms and all mseed formats. This program can be used as alternative to converting data with wavetool, advantage is that SEISAN2MSEED supports STEIM2 compression.
Source code can be found at https://seiscode.iris.washington.edu/projects/seisan2mseed
SYNOPSIS
seisan2mseed [options] file1 [file2 file3 ...]
Seisan2mseed converts SeisAn waveform data files to Mini-SEED. One or more input files may be specified on the command line. If an input file name is prefixed with an ‘@’ character or explicitly named ‘filenr.lis’ the file is assumed to contain a list of input data files, see LIST FILES below. The default translation of SeisAn components to SEED channel codes is as follows: a 3 character SEED channel is composed of the first, second and fourth characters of the component; furthermore if the second character is a space and the first and fourth are not spaces an ‘H’ is substituted for the 2nd character (i.e. ‘S_Z’ → ‘SHZ’). The default SEED location code is ‘00’, if the third character of the SeisAn component is not a space it will be placed in the first character of the SEED location code. Other translations may be explicitly specified using the -T command line option. If the input file name is a standard SeisAn
file name the default output file name will be the same with the 'S' at character 19 replaced by an 'M'. Otherwise the output file name is the input file name with a "_.MSEED" suffix. The output data may be redirected to a single file or stdout using the -o option.

OPTIONS

"-V"
Print program version and exit.

"-h"
Print program usage and exit.

"-v"
Be more verbose. This flag can be used multiple times ("-v -v" or ";-vv") for more verbosity.

"-S"
Include SEED blockette 100 in each output record with the sample rate in floating point format. The basic format for storing sample rates in SEED data records is a rational approximation (numerator/denominator). Precision will be lost if a given sample rate cannot be well approximated. This option should be used in those cases.

"-B"
Buffer all input data into memory before packing it into Mini-SEED records. The host computer must have enough memory to store all of the data. By default the program will flush its data buffers after each input block is read. An output file must be specified with the -o option when using this option.

"-n Inetcode P"
Specify the SEED network code to use, if not specified the network code will be blank. It is highly recommended to specify a network code.

"-l Iloccode P"
Specify the SEED location code to use, if not specified the location code will be blank.

"-r Ibytes P"
Specify the Mini-SEED record length in Ibytes P, default is 4096.

"-e Iencoding P"
Specify the Mini-SEED data encoding format, default is 11 (Stein-2 compression). Other supported encoding formats include 10 (Stein-1 compression), 1 (16-bit 3 integers) and 3 (32-bit integers). The 16-bit integers encoding should only be used if all data samples can be represented in 16 bits.

"-b Ibyteorder P"
Specify the Mini-SEED byte order, default is 1 (big-endian or most significant byte first). The other option is 0 (little-endian or least significant byte first). It is highly recommended to always create big-endian SEED.

"-o Ioutfile P"
Write all Mini-SEED records to Ioutfile P, if Ioutfile P is a single dash (-) then all Mini-SEED output will go to stdout. All diagnostic output from the program is written to stderr and should never get mixed with data going to stdout.

"-T Icomp=chan P"
Specify an explicit SeisAn component to SEED channel mapping, this option may be used several times (e.g. "-T SBIZ=SHZ -T SBIN=SHN -T SBIE=SHE"). Spaces in components must be quoted, i.e. "-T 'S Z'=SHZ".

LIST FILES
If an input file is prefixed with an '@' character the file is assumed to contain a list of file for input. As a special case an input file named 'filenr.lis' is always assumed to be a list file. Multiple list files can be combined with multiple input files on the command line.

The last, space separated field on each line is assumed to be the file name to be read. This accommodates both simple text, with one file per line, or the formats created by the SeisAn dirf command (filenr.lis).

An example of a simple text list:
2003-06-20-0643-41S.EDI
2005-07-23-1452-04S.CER

An example of an equivalent list in the dirf (filenr.lis) format:

- # 1 2003-06-20-0643-41S.EDI
- # 2 2005-07-23-1452-04S.CER

SILSEI: all, chan. def. no, resp no
Conversion from the Icelandic SIL system to SEISAN. Only conversion from ASCII files.

SISSEI: Sismalp to SEISAN
- all, chan. def. yes, resp yes
The program converts from Sismalp to SEISAN. Sismalp is a French field recording system. The input consists of 2 files per event, a header file and a data file. It is assumed that the Sismalp ndx files have the same file name as the header file except for the file extension. It is also assumed that the file names are 12 characters long.

SUDSEI: PCSUDS to SEISAN
- PC, chan. def. yes, resp yes
The program converts from PCSUDS to SEISAN. This is done by first running the program SUD2ASC (included) and then converting to SEISAN. The SUD2ASC program and test data was supplied by REFTEK through the distribution of PC-SUDS Utilities by Banfill [1996].

TERSEI: Terra ASCII to SEISAN
- all, chan. def. yes, resp yes
Program converts from Terra Technology ASCII files to SEISAN. Only tested with 1-3 channel files

VELSEI: Velest CNV to Nordic

An example of the input format is seen below:

```
191213 043 55.85 41.4825N 19.6017E 26.19 -99.00 98 0.09
AB18P1 3.99AB26P0 4.07AB06P0 4.08AB30P0 4.22AB10P0 4.74AB10S0 9.01
AB17P0 4.78AB09P0 4.85AB09S1 9.21AB03P1 5.95AB03S2 11.12AB02P0 6.02
AB01P1 6.27AB13P1 7.12
191213 131 28.93 41.4204N 19.5849E 10.39 -99.00 93 0.08
AB26P1 1.91AB18P3 2.04AB30P0 2.05AB06P0 2.18AB09P1 2.22AB10P0 2.26
AB17P0 2.54AB02P1 3.66AB03P1 3.68AB01P1 3.93AB13P2 4.54
```

WGSSEI to SEISAN: all, chan. def. yes, resp yes
Program converts from WGSN files to SEISAN. The format is used on IRIS stations as processing format. Little tested.

WAVASC: ASCII file from any WAV file

Program to convert any seisan waveform file to a 2 column ASCII file. One file will be made per channel as well as one file for all the converted data. The start of the converted file looks as follows:

```
COCO BH E 10 II 2007 928 1345 31.440 42953 40.000
31.440 1874.00
31.465 1787.00
```
where the header has station, component, location, network, time, number of samples and sample rate. The following times are referred to the minute in the header.

the channel output filename is like

2007-09-28-1345-31A.COOCO_BH_E

where a stands for ASCII

The input can be an s-file where all waveform files in s-file will be used or it can be a waveform file or a filenr.lis of waveform files.

19.1 Nordic to QuakeML converter

Author Christian Rønnevik

This application is a conversion tool between the Nordic format and QuakeML. The tool is preconfigured to reading (s-files) and producing earthquake event data (event parameters) in QuakeML. By configuring arguments the tool can also convert QuakeML data into a collection of s-files from both a XML file and a web-service.

1. How to run the application
2. Configure ID generation
3. Configure to convert from QuakeML to nordic
4. Arguments

Works only with old Nordic format.

How to run the application

Make sure that you have java installed. Check by typing ”java -version” into the terminal.

Run without options:
To run the application from Nordic file to QML, type the following command into the terminal:

nor2qml file where file is Nordic input file or from QML to Nordic
qml2nor file, where file is QML file

If you wish to use the application in isolation outside the SEISAN environment, simple copy the ”nor2qml.jar” file into your desired place and run the application by typing ”java -jar nor2qml.jar –input=”¡path to file or folder¡” –output=”¡path to file¡” into the terminal. If the nor2qml.jar file is in the same folder as the file you wish to convert, you can omit both input and output arguments. Then the converter will identify all s-files within that folder and make a output file in the same directory with the filename ”quakeml.xml”.

Input
The application has a built-in file scanner, which is designed to identifying s-files within in SEISAN
catalog structures. Hence, a folder will only be scanned if the folder name follows the convention of a
catalog name with underscore ("_") at the end, for example the catalog name "NNSN_". The file scanner
will also search through folders with a folder name of numbers, which is true for the catalog’s subsequent
folders of year and month. Examples of valid input for folders are:

" Folder (whole catalog): "<path to seisan>/SEISMO/REA/NNSN_"
" Folder (specific year): "<path to seisan>/SEISMO/REA/NNSN_/2014"
" Folder (specific month): "<path to seisan>/SEISMO/REA/NNSN_/2014/09"

During file scan, only files that has the extension starting with "S" will be read. However, if the given
input path is pointing at a single file, the application can read any file regardless of extension or filename.
Hence, following examples of file input can be used:

" File (particular file): "<path to seisan>/SEISMO/REA/NNSN_/2014/09/01-2050-05L.S201409"
" File (particular file): "<path to any folder>/testfile.out"

Output
The output argument has a default value of the path where the application (jar file) is located. If you
are running the converter through the seisan structure it is recommended to specify another path for the
output. The output value should the complete path with a filename and extension at the end of the path.
If converting from Nordic to QuakeML, please give the "xml" extension. If converting from QuakeML,
you are more free to chose extension, but as a guideline you can use the ".out" extension. Example for
running the application

nor2qml --input="/home/SEISMO/REA/NNSN_/2014/09" -output="/home/my-name/qml201409.xml"

Configure ID generation
The application has a build-in ID generator that follows the QuakeML standard of resource identifiers.
As written in the QuakeML documentation, the identifiers follow this generic form:

[smi|quakeml]:<authority-id>/<resource-id>[#<local-id>]

More explicitly the id follow this schema pattern:

(smii | quakeml) : [ \w\d ] [ \w\d ?\.|\?\|\|\| ] { 2 , } / [ \w\d ?\.|\?\|\|\| ] [ \w\d ?\.|\?\|\|\|\| ]

The two first part of this identifier is configurable in the application by the arguments prefix and agency.
To see a list and description of all the available arguments, please see the arguments section below.
Prefix
The prefix argument takes a sting value that should be either "smi" or "quakeml". To apply this argu-
ment, simple attach --prefix="smi" or --prefix="quakeml" when running the application. "smi" stands
for seismological meta-information. Agency Similarly, the authority-id can be configured by attaching
--agency="¡your authority-id¿", which should be at least 3 characters with the first character as alphabet-
numeric, while subsequent characters can be alphanumeric or from the following list:

-, _, ., "", *, ', (, ).

The agency-id should follow this form:
19.1. NORDIC TO QUAKEML CONVERTER

\langle\text{top-level domain}\rangle.\langle\text{organisation/institution}\rangle[.\langle\text{sub-unit of organisation}\rangle]

An example can be \text{"no.uib.nnsn"}, where \text{"no"} (Norway) is the top-level country domain, \text{"uib"} (University of Bergen) is the organisation and \text{"nnsn"} (Norwegian National Seismic Network) is the sub-unit of the organisation. Example with ID generation Running the application for an s-file within the NNSN catalog could then look like this:

\text{nor2qml --input="/SEISMO/REA/NNSN_/2014/09" --output="/home/my-name/qml201409.xml"  --prefix="smi"}

Configure to convert from QuakeML to Nordic
Conversion from QuakeML to the Nordic format can be achieved from two different sources, a XML file or a web-service.

QuakeML file to Nordic
To convert a QuakeML file to the Nordic format, three arguments is required: convert, input and output arguments. First, an input is needed and for file conversion the tool is configured to read single XML files. Hence, attach \text{--input="\text{path to file}" } to the nor2qml command. Output is also given by a path to the file with chosen extension. Therefore, attach \text{--output="\text{path to file}" } to the command. By default, the convert argument is set to \text{"q"} (meaning QuakeML), specifying the format s-files should be converted to. In order to convert to Nordic, type attach \text{--convert="s" } to the command. The full command would then look like:

\text{nor2qml --input="\text{path to file}" --output="sfiles.out" --convert="s"}

QuakeML web-service to Nordic
Configuring to read from a QuakeML web-service requires four arguments: output, convert, source and url. Both output and convert are set in a similar fashion as for converting a quakeml file. The source parameter is telling the application that it should fetch data from an external web-service. An internet connection is thus required to make use of this command. An url to the web-service is also required and should be given in the url parameter. The full command would then look like:

\text{nor2qml --output="sfiles.out" --convert="s" --source="ws" --url="\text{path to web-service}"}

Arguments
General Arguments
### Chapter 19. File Conversion and Modification Programs

<table>
<thead>
<tr>
<th>Key</th>
<th>Value type</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>convert</td>
<td>&quot;q&quot; or &quot;s&quot; (type String)</td>
<td>&quot;q&quot;</td>
<td>Specify to which format the files should be converted to. &quot;q&quot; means convert from s-files to QuakeML, while &quot;s&quot; means convert QuakeML to s-files.</td>
</tr>
<tr>
<td>input</td>
<td>Path (type String).</td>
<td>Current path</td>
<td>Select where the input files are. If no path is specified, the program will search the current folder for files to be converted.</td>
</tr>
<tr>
<td>output</td>
<td>Path (type String).</td>
<td>Current path</td>
<td>Select where the converted file should be placed. If no path is specified, the program will create a single file in the current folder with the filename and extension to either &quot;quakeml.xml&quot; or &quot;sfile.out&quot; based on the conversion type.</td>
</tr>
</tbody>
</table>

#### QuakeML specific arguments

<table>
<thead>
<tr>
<th>Key</th>
<th>Value type</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>&quot;1.2&quot; or &quot;2.0&quot;</td>
<td>&quot;1.2&quot;</td>
<td>Specifying the QuakeML version output.</td>
</tr>
<tr>
<td>prefix</td>
<td>String (that should be &quot;smi&quot; or &quot;quakeml&quot;)</td>
<td>&quot;smi&quot;</td>
<td>Specifying the prefix of generation of IDs</td>
</tr>
<tr>
<td>agency</td>
<td>String</td>
<td>&quot;authorityid&quot;</td>
<td>Specifying the agency for generation of IDs (e.g. &quot;no.msn&quot;)</td>
</tr>
<tr>
<td>source</td>
<td>String (that should be &quot;file&quot; or &quot;ws&quot;)</td>
<td>&quot;file&quot;</td>
<td>Specify if the source is a file or a web-service (ws)</td>
</tr>
<tr>
<td>url</td>
<td>String (mandatory if source is ws). The value must be enclosed with &quot;.&quot;.</td>
<td></td>
<td>Specifying the url if the source is a web-service</td>
</tr>
</tbody>
</table>

#### References

Chapter 20

Signal processing programs

The SAC software is interfaced to SEISAN and can be started directly from EEV. This is done since SAC has functions that SEISAN does not have. It only operates on Unix.

20.1 SAC2000

SAC2000 (seismic analysis code) is currently developed by Lee Minner and Peter Goldstein [Goldstein, 1999]. SAC is not distributed with SEISAN, information on SAC can be obtained from the SAC homepage [http://www-ep.es.llnl.gov/www-ep/esd/seismic/sac.html]. The main features of SAC include general arithmetic operations, Fourier transforms, three spectral estimation techniques, IIR and FIR filtering, signal stacking, decimation, interpolation, correlation, and seismic phase picking. SAC also contains an extensive graphics capability. With SAC it is possible to write macros, which helps to process large amounts of data. The SAC format is used in several research oriented programs. SAC can be started from EEV using the command ‘sac’. EEV will start the WAVETOOL program to convert the data to SAC and then execute the command sac. In case your sac executable is called sac2000, it is necessary to rename it (to sac) or alternatively to create a link in either the SEISAN PRO directory or the SAC bin directory. This is done for example by the command:

```
ln -s /sac/bin/sac2000 /sac/bin/sac
```

Since the SAC format is a single trace format, the SEISAN multichannel files are split into single trace files. The station and component names are included in the file name and the suffix ‘SAC’ is added to all SAC files. For both systems, waveform data can be converted to the respective format outside EEV using WAVETOOL, GSESEI or SACSEI, and the programs can be started without using EEV.
Chapter 21

Automatic routines in SEISAN

The section describes the routines in SEISAN that can be used for analysing data automatic.

21.1 AUTOPIC and AUTO

AUTOPIC is a tool to automatically pick phases on events registered into the database. AUTOPICK can also be used for many events in connection with program AUTO, see below.

If an event file (S-file) has any readings, the AUTOPIC program will not reread in order to not destroy old picks. The automatic readings in the file are marked with an A after the weight column to indicate automatic pick. Each pick is evaluated by using the signal to noise ratio and an indication of the quality is given with the weight. The program will run on all waveform files given in an S-file. Each time the program runs, there is a file called autopic.out containing information about the run. If there are any 3-component stations, an azimuth will also be calculated, and the S-phase will be more reliable. The AUTOPIC program can also be used from EEV by typing Z (will run program AUTOPIC). When it is used from EEV, there is always an output in the S-file, which will be grouped at the bottom of the file, making it possible to compare manual and automatic readings. THE S-FILE MUST THEN BE EDITED MANUALLY IN ORDER TO REMOVE DOUBLE READINGS. The program requires an input parameter file in the working directory or DAT with the name AUTOPIC.INP. The program will first look in the working directory. The parameters in that file are explained below. NOTE: The file is formatted, data must be in columns exactly as shown and no tabs must be used. The program uses a 4-pole filter running one way. This might result in phases being picked a bit late. However, it seems more accurate than the earlier version where the filter run both ways and picks were often far too early. The program is made mainly by Bent Ruud. For more information about how it works, see Ruud et al. [1988]; Ruud and Husebye [1992].

Description of parameters:

% Input parameters common to all filters:
%
% LWIND : used to define step length (DELTA=WINDOW/LWIND)
% ISHIFT : defines time shift between STA and LTA window (ISHIFT*DELTA)
% Delay for LTA window (15 * 0.1) ==> 1.5 sec. after STA window.
% ISIGMA : defines fall off rate of LTA window (larger values longer windows)
% COHMIN : Polarization threshold.
% Minimum coherence (see thresh_1 and thresh_2)
% NDMin : Minimum number of consecutive triggered windows in a detection
% SVELO : S wave velocity of the medium below the station (used for 3 comp)
% NFILT : number of filters
% CRAT : Ratio for calculation of coda duration (range 1 4)
% LWIN : Window used in coda duration routines (range 20 50 seconds)
% THRES : Quality threshold (range 2 5). Used on the maximum to average
% amplitude ratio in order to sort out the most noisy traces.
%
% Input parameters defined for each filter:
%
% WINDOW : length of the moving time window (sec)
% F1 : lower cutoff frequency (Hz) of band pass filter
% F2 : higher cutoff frequency (Hz) of band pass filter
% THRSH1 : STA/LTA threshold for polarized signals
% THRSH2 : STA/LTA threshold for unpolarized signals
% If coherence > cohmin then detection is made on thresh_1
% If coherence < cohmin then detection is made on thresh_2
%
% Output parameters:
%
% D : day of year
% H : hour
% M : minute
% SEC : second
% DUR : duration, i.e. time in detection state (sec)
% FRQ : centre frequency of filter giving the best detection (Hz)
% SNR : signal to noise ratio (SNR=STA/LTA)
% STA : short time average (root mean square of amplitude)
% NT : total number of triggered time windows in the detection
% NH : number of windows with best SNR on one of the horizontal comp.
% NV : number of windows with best SNR on the vertical comp.
% NC : number of windows with acceptable polarization
% Q : quality class, 1(best) 4(worst)
% PS : P/S wave discriminator, 0(S) 10(P)
% AZI : backazimuth in degrees measured from North through East
% DA : variability in azimuth (deg)
% VEL : apparent velocity (km/s)
% DV : variability in apparent velocity (km/s)
%
% Note: azimuth and apparent velocity calculations are based on the
% assumption of P wave, so that these variables should be
% neglected for S waves.
Example of input file AUTOPIC.INP

% This is the parameterfile needed by program: --- PREPROCESS ---
%
% The following rules apply:
% 1. All lines with % in the first column are comment lines
% 2. Lines with a blank in column 1 are read for fixed parameters.
% 3. All lines starting with "filter_x", where x is a number,
%    are read for filter variable parameters
% 4. All lines with * in the first column are read for stations to process
% 5. A brief explanation of all parameters is given in preprocess.inf
%
% FIXED PARAMETERS THAT ARE USED THROUGHOUT THE PROGRAM
% ----------------------------------------------------------------------
% Lwind  Ishift  Isigma  Cohmin  Ndmin  Svelo  Nfilt  Crat  Lwin  Thres
% ! ! ! ! ! ! ! ! ! !
% ----------------------------------------------------------------------
% 4.0 30.0 06.0 0.1 3.0 2.75 4.0 1.6 30.0 3.0
%
% PARAMETERS THAT ARE FILTER DEPENDANT
% ----------------------------------------------------------------------
% Filter_nr  Window  F1  F2  Thrsh1  Thrsh2
% ----------------------------------------------------------------------
% filter_1  0.8  2.0  4.0  2.30  3.0
% filter_2  0.6  5.0 10.0  2.30  3.00
% filter_3  0.4  8.0 16.0  2.30  3.00
% filter_4  2.0  0.5  2.0  4.0   5.0
%
% STATIONS TO USE IN THE PROCESSING
% ----------------------------------------------------------------------
% *SUE  S  Z  3 component
% *BER  S  Z
% *HYA  S  Z
% *KMY  S  Z  3 component
% *ODD1 S  Z  3 component
% *BLS5 S  Z  3 component
% *ESG  S  Z
% *EGD  S  Z
% *KTK1 S  Z  3 component
% *NSS  S  Z  3 component
% *MOL  S  Z  3 component
% *MOL  S  A  3 component
% *JNW  S  Z
% *JNE  S  Z
% *FRO  S  Z
% *JMI  S  Z  3 component
% *ASK1 S  Z  3 component
*ASK  S  Z 3 component
*MOR7  S  Z 3 component
*MOR8  S  Z 3 component
*LOF  S  Z 3 component
*LOF  A  Z 3 component
*OSG  S  Z 3 component
*TRO  S  Z 3 component
*FOO  S  Z 3 component
*ALVN  S  Z 3 component
*UGA  S  Z 3 component
*ENT  A  Z 3 component

AUTO

A Program for automatic processing: Phase picking, location, magnitude and fault plane solution determination.

This program runs the picking program AUTOPHASE first (optionally AUTOPHASE), locates events with or without outlier removal with HYP and then determines magnitudes with AUTOMAG. Automatic amplitudes for fault plane solutions can be done optionally with AUTORATIO. Fault plane solutions can also be done optionally with FPFIT or HASH. Note velocity is the hardwired option. The default is to do AUTOPHASE, HYP with outlier removal and AUTOMAG using default parameters. Individual steps, like doing magnitudes, can optionally be deselected and some non default parameters for AUTOMAG can be selected. All changes are made using arguments, see list below. Thus the program can also be used to do any one of the operations. However to do only e.g. magnitudes, a more rational choice would be to use AUTOMAG only. Similarly for doing only locations, it is more logical to use HYP. The input can be all SEISAN types: a file, a data base or an index file. Outlier removal by HYP can be deselected. An alternative to outlier removal by HYP is to use the HYP option for residual weighting. For more detail of the individual programs, see program descriptions elsewhere in SEISAN manual. There are always two output files: auto.out: all events with the final results with updated values of location and magnitude. auto.log: a summary of what has been done to each event. If input is from a data base or index file, the results are also written back to the data base, overwriting what was there from before. The ID line is not updated by AUTO. When picking automatically, a few bad picks can throw the solution off so for a local event, the distance might wrongly be very large. It is then important that the range used for distance weighting in HYP is large so the initial wrong location can be made and the outlier rejection gets a chance to eliminate bad picks. Doing fault plane solution is only recommend with very good data. It is then important to use at least the default values set for both minimum number of polarities and maximum az gap. Note the gap is not the gap as in a hypocenter solutions but the gap when polarities are plotted on the projection on the focal sphere.

See list of argument below

Examples of argument use:

auto : pick phases, locate and do magnitudes
auto l m : only pick phases with AUTOPHASE
auto j l m : only pick phases with AUTOPIC
auto i l : only do magnitudes
auto s 30 : all defaults except that spectral window for magnitude is 30 s
auto i m i f : only do fault plane solutions with FPFIT
auto i m i ar h f: automatic amplitudes and fault plane solution with HASH
Writing auto help will give all the possible arguments:

c:\seismo\PRO>auto help
Write auto help to get list of arguments. If no arguments, all defaults are used. This is autophase, location with outlier rejection and ML and spectral Mw

s xxx : do spectrum for Mw, xxx is window length, default is 20 s. if 0, no spectrum. Default is spectrum

w xxx : amplitude for Ml, xxx is window length
if zero, no amplitude. Default is do amplitude
default window length is 50 s

n or e: use N or E component for magnitude, respectively.
  Default is Z

p : use P for Mw, default is S

l : do not locate, def. is to locate with rejection
  if magnitudes are done, location is also done
  after determination of amps and spectra in order
  to update magnitudes on header line

r : do not remove outliers when locating

m : no magnitude, default is to do magnitude

i : no phase pick, default is to do phase pick

j : if phase pick, use autopic, default is autophase

ar : do autoratio, default is not to do

af xx xx : filter for autoratio, default is 2-4 Hz

ad xx : max distance for autoratio, def 100km

at xx : time window for autoratio, def 2 s

ag xx : ground motion for autoratio, def 0 for none

az : if given, use z for s in autoratio

f : do fault plane solution with fpfit, def. is not

h : do fault plane solution with hash, def. is not
n : minimum number of polarities for fps, default 10

g : maximum gap for fps, default 180

When using amplitude ratios for fps, velocity is the harwired default.

21.2 AUTOPHASE, automatic phase picking

By T. Uttheim terje.utheim@geo.uib.no

The AUTOPHASE program is part of the RTQUAKE real-time earthquake detection [Utheim et al., 2014]. The AUTOPHASE (hereafter AP) is a stand-alone phase-picker routine that is used for automatic phase-picking in RTQUAKE and can also be used as an option in SEISAN from the EEV program. The routine is based on the FilterPicker algorithm [Lomax et al., 2012], but with several modifications and additions. However, the original picking algorithm is unchanged.

The FilterPicker (hereafter FP) implementation in both RTQUAKE and SEISAN checks each component of a waveform for possible phase-picks. All channels in the waveform file are checked and there is no possibility to unselect channels. Each component may have several suggested phases given by the time of occurrence. Each component may also have a suggested first movement indicated for each pick. There can thus be many picks for each channel.

The following rules apply for selecting P and S phases among the suggested picks:

For one station there will only be one P and one S-phase selected.
P is only accepted on vertical components.
S is only accepted on vertical components if the station does not have horizontal components so S is in general only accepted on horizontal components. If several picks are found on a horizontal component, it is assumed that the first is P. The S is then selected on the first of the following picks that has a time-difference greater than a default value MINDIFF (1.5 seconds) set in the program. This is to avoid picking secondary P phases as S phases. This rule is also used if picking S on Z-components.
If S-phases are found on two different horizontal channels, the first arriving one is used.
If only one pick on a horizontal component, no S is selected.
First movement are included in the S-file only for vertical channels.
If the event is defined as distant (D) in the SEISAN S-file, only P phases are picked.
S-phases are given a weight of 3.

Running AUTOPHASE AP is started from within the SEISAN EEV program with the command 'ap'. This command converts all associated waveform files (can be different formats) to one MiniSeed file seed.out, which is then used as input to AUTOPHASE. The S-file and corresponding waveform files are used as input to the AP. All old readings are deleted and replaced by the new readings from AP. All phases are marked with A for automatic. AUTOPHASE can also be operated outside EEV with the command:

Autophase -sfile xxx -wavefile yyy

where xxx is a single event S-file and yyy is one waveform file in MiniSeed format. AP can only work
with one S-file.

AUTOPHASE also has other options not used in SEISAN, see RTQUAKE manual:


Parameters

AUTOPHASE has no parameter file since it seems that the default parameter work will in most cases.

Default FilterPicker parameters are set at the top in the source program under the comment:

"Default FilterPicker parameters” and can be changed by the user. See M. Vassallo and Lomax [2012] for documentation. The program must be recompiled.

#define Picker_FW 300.0 Longest period for a set of filtered signals from the differential signal of the raw broadband input trace (300 multiplied by sample rate)

#define Picker_LTW 500.0 Time scale used for accumulating time-averaged statistics of the input raw signal (500 multiplied by sample rate)

#define Picker_TH1 10.0 Trigger threshold used for event declaration. A trigger is declared when the summary CF (characteristic function) exceeds Picker_TH1

#define Picker_TH2 10.0 A pick is declared if and when, within a window of predefined time width, Picker_TUP after trigger time, the integral summary CF (characteristic function) exceeds the value: Picker_TH2 multiplied by Picker_TUP (see below).

#define Picker_TUP 20.0 Time window used for pick validation (20 multiplied by sample rate).

Default time-difference MINDEF is set to:

MINDEF = 1.5

Coda is computed for vertical components only, and only for events marked local (L) or regional (R) in the s-file. Computation of coda can be turned off by changing the SETCODA switch in the source code to zero, and recompiling. Default is SETCODA=1. The coda is computed by comparing a long-term average of the signal before the P phase with long-term averages after the P-phase divided by a factor.

Example: The AUTOPHASE picker is started from EEV with the command: ap

# 13 9 Dec 2015 08:03 54 LM-22.259 -66.832 0.0 N 1.2 2.3WBER 11 ? ap

The waveform file is converted to miniseed file: seed.out

wavetool -sfile /home/seismo/snew/REA/TST__/2015/12/09-0804-18R.S201512 -format MSEED -wav_out_file seed.out

Number of wav-files 1
Number of wav-files present 1
/home/seismo/snew/WAV/TST__/2015/12/2015 MINISEED
Total number of channels available: 42
Total duration: 323.599976

Output waveform file name is seed.out
Here the autophaser routine is started with s-file and seed.out file as input:
autophase -sfile /home/seismo/snew/REA/TST__/2015/12/09-0804-18R.S201512 -wavefile seed.out
SEISAN_TOP.......................: /home/seismo/snew
wfilename: seed.out
All picks found are listed below in the format:

STATION-NAME DATE TIME TIME-DIFFERENCE TO NEXT PICK ON SAME COMPONENT

CX_PB16__00_BHZ 09/12/15 08:05:04.499 (1 pick only)
Figure 21.1: Examples of automatic picks on a "good" event.

CX_PB11__00_BHZ 09/12/15 08:04:49.150 39.85 (2 picks, time-diff. to 2. pick: 39.85 seconds)
CX_PB11__00_BHN 09/12/15 08:04:49.349 39.30
CX_PB11__00_BHE 09/12/15 08:04:49.349 39.55
CX_PB08__00_BHZ 09/12/15 08:04:43.769
CX_PB08__00_BHN 09/12/15 08:04:43.819 35.35
CX_PB08__00_BHE 09/12/15 08:04:43.919 35.40
CX_PB01__00_BHZ 09/12/15 08:04:38.269
CX_PB01__00_BHN 09/12/15 08:04:38.569 31.25
CX_PB01__00_BHE 09/12/15 08:04:38.569 31.95
CX_PB07__00_BHZ 09/12/15 08:04:39.019
CX_PB07__00_BHN 09/12/15 08:04:38.969
CX_PB07__00_BHE 09/12/15 08:04:39.069
CX_PB09__00_BHZ 09/12/15 08:04:33.650
CX_PB09__00_BHN 09/12/15 08:04:33.999
CX_PB09__00_BHE 09/12/15 08:04:33.799
CX_PB03__00_BHZ 09/12/15 08:04:37.569
CX_PB03__00_BHN 09/12/15 08:04:37.469
CX_PB03__00_BHE 09/12/15 08:04:37.469
CX_PB04__00_BHZ 09/12/15 08:04:41.369
CX_PB04__00_BHN 09/12/15 08:04:41.669
CX_PB04__00_BHE 09/12/15 08:04:41.619 0.90
CX_PB06__00_BHZ 09/12/15 08:04:36.519
CX_PB06__00_BHN 09/12/15 08:04:36.919 30.15
CX_PB06__00_BHE 09/12/15 08:04:36.769
21.2. AUTOPHASE, AUTOMATIC PHASE PICKING

CX_PB05__00_BHZ 09/12/15 08:04:42.619
CX_PB05__00_BHN 09/12/15 08:04:43.069
CX_PB05__00_BHE 09/12/15 08:04:42.769
CX_PB15__00_BHZ 09/12/15 08:04:37.869 1.20 30.50 (3 picks, and 2 time-differences)
CX_PB15__00_BHN 09/12/15 08:04:37.669 31.65
CX_PB15__00_BHE 09/12/15 08:04:38.169 31.30
CX_PB10__00_BHE 09/12/15 08:04:48.500

SELECTED PICKS:

Below the selected picks from the picks above. As seen below, there are picks on both horizontal components on stations PB11, PB08, PB01 and PB15:

0 PB16 BHZ P 09/12/15 08:05:04.499
6 PB11 BHZ P 09/12/15 08:04:49.150
7 PB11 BHN S 09/12/15 08:05:28.649
8 PB11 BHE S 09/12/15 08:05:28.899
9 PB08 BHZ P 09/12/15 08:04:43.769 C
10 PB08 BHN S 09/12/15 08:05:19.169
11 PB08 BHE S 09/12/15 08:05:19.319
12 PB01 BHZ P 09/12/15 08:04:38.269
13 PB01 BHN S 09/12/15 08:05:09.819
14 PB01 BHE S 09/12/15 08:05:10.519
15 PB07 BHZ P 09/12/15 08:04:39.019
18 PB09 BHZ P 09/12/15 08:04:33.650 C
21 PB03 BHZ P 09/12/15 08:04:37.569
24 PB04 BHZ P 09/12/15 08:04:41.369
27 PB06 BHZ P 09/12/15 08:04:36.519
28 PB06 BHN S 09/12/15 08:05:07.069
30 PB05 BHZ P 09/12/15 08:04:42.619
33 PB15 BHZ P 09/12/15 08:04:37.869
34 PB15 BHN S 09/12/15 08:05:09.319
35 PB15 BHE S 09/12/15 08:05:09.469

In the next step of selection the first of the 2 S-phases on the horizontal components are selected while the last are deleted:

0 PB16 BHZ P 09/12/15 08:05:04.499
6 PB11 BHZ P 09/12/15 08:04:49.150
7 PB11 BHN S 09/12/15 08:05:28.649
9 PB08 BHZ P 09/12/15 08:04:43.769 C
10 PB08 BHN S 09/12/15 08:05:19.169
12 PB01 BHZ P 09/12/15 08:04:38.269
13 PB01 BHN S 09/12/15 08:05:09.819
15 PB07 BHZ P 09/12/15 08:04:39.019
18 PB09 BHZ P 09/12/15 08:04:33.650 C
21 PB03 BHZ P 09/12/15 08:04:37.569
24 PB04 BHZ P 09/12/15 08:04:41.369
27 PB06 BHZ P 09/12/15 08:04:36.519
28 PB06 BHN S 09/12/15 08:05:07.069
30 PB05 BHZ P 09/12/15 08:04:42.619
The autophase routine will also indicate first movement of p-phases when found. This can be seen on station PB08 BZ and PB09 BZ below:

<table>
<thead>
<tr>
<th>Time</th>
<th>Station</th>
<th>Location</th>
<th>Latitude</th>
<th>Longitude</th>
<th>Bearing</th>
<th>Arrival Time</th>
<th>Distance</th>
<th>Velocity</th>
</tr>
</thead>
<tbody>
<tr>
<td>00:03</td>
<td>PB08 BZ</td>
<td>22.259</td>
<td>-66.832</td>
<td>0.0</td>
<td>BER</td>
<td>11</td>
<td>2.3</td>
<td>0.5</td>
</tr>
<tr>
<td>00:04</td>
<td>PB09 BZ</td>
<td>22.259</td>
<td>-66.832</td>
<td>0.0</td>
<td>BER</td>
<td>11</td>
<td>2.3</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Create s-file: /home/seismo/snew/REA/TST__/2015/12/09-0804-18R.S201512

# 13 9 Dec 2015 08:03 54 LM-22.259 -66.832 0.0 N 1.2 2.3WBER 11 ?
AUTOSIG is a program to perform some automatic processing. The program includes routines for
P-phase picking, determination of signal duration, amplitude determination, determination of spectral
parameters [Ottemöller and Havskov, 2003] and determination of distance type (local, teleseismic, noise).
The program can still do with improvement. The input to the program can be either a parametric Nordic
file (with one or several events) or waveform files. In both these cases, the output is written to the
autosig.out file. Additional output files are autosig.trace and autosig.err, which will help to find
potential problems. Alternatively, the program can also be started from EEV; the output is then directly
written to the S-file. The input parameters are defined in the file autosig.par, which is located either
in the DAT or the working directory.

Following are descriptions of the automatic processing routines:

P-phase picking:
The phase picking is based on changes in the STA/LTA ratio. A band pass filter can be specified. The
routine gives options to enhance the changes in the signal before computing the STA/LTA ratio. It is
recommended to first remove the linear trend and then to compute the characteristic function which is
given by $y^2 + k(\frac{dy}{dt})^2$, which enhances changes in both amplitude and frequency content. Then
the STA/LTA ratio is computed to detect changes in the signal. The routine can also compute the
squared STA/LTA. When a change is detected (STA/LTA ratio above trigger level), it is tested whether
the signal spectral amplitudes are significantly higher (factor of 2 in amplitude) than the pre-signal noise
spectral amplitudes. This is done to avoid triggering on spikes.

Signal duration:
The signal duration is determined by comparing the signal amplitudes with the amplitudes of the pre-
signal noise. The duration is determined by the point from which the ratio of these amplitudes is lower
than a given value. A filter is applied if specified in the parameter file.

Amplitude:
Routine finds maximum amplitude between two peaks.

Spectral parameters:
The routine computes the displacement amplitude spectrum for P or S waves (see section 8.9) and, using
either a converging grid search or a genetic algorithm determines the seismic moment and the corner
frequency by minimizing the difference between observed and synthetic source spectra. The frequency
band is determined by comparison with the pre-signal spectrum. The grid search is generally more cost
effective and produces better results. The method is described in [Ottemöller and Havskov, 2003]. The
displacement spectrum is corrected for geometrical spreading and attenuation (both along the travel path
and near surface). Therefore, the hypocentral distance has to be known. The time domain window for
extracting the data from the trace can be given by either a group velocity ($V_g=\frac{d}{d'\text{fick}}$) window or a fixed window in seconds around the phase pick.

Distance type:
The routine determines whether the signal is from a local or teleseismic event, or noise. If signal spectral
amplitudes are not significantly higher than pre-signal noise amplitudes, it is assumed that the signal
is noise. Otherwise the amplitudes at two selected frequencies given by ‘DIST FREQ SELECT’ are
compared, the rules are ($f_1<f_2$):

- Spec signal amp($f_1$) - Spec noise amp($f_1$) > Spec signal amp($f_2$) - Spec noise amp($f_2$): teleseismic
- Spec signal amp($f_2$) > Spec noise amp($f_2$): local

There are a few command line options that can be used to run autosig in non-interactive mode, syntax is
autosig -infile <filename> [-spec on/off -phase on/off -clear on/off]

where

-spec on/off: determine spectral parameters if option given
-clear on/off: remove phases from input S-file before start if option given
-phase on/off: detect phases if option given
-infile <file>: give name of input file, either S-file or waveform file
-help: get help

Note: When running the program the first time and the hypocenter location is not known, the determination of spectral parameters is not done. To run the determination of spectral parameters, the hypocenter location has to be given in the S-file.

The meaning of most parameters in the parameter file is clear from the keyword. The spectral parameters are as described in the MULPLT section. Other parameters that need explanation are:

**AUTO PHASE, AUTO SPECTRUM and AUTO AMPLITUDE**: Logical flag to activate phase picking, spectral analysis and amplitude reading, respectively (1. for true)

**GA POPULATION SIZE**: Number of elements in the population, used only if SEARCH ALGORITHM is 1.

**GA GENERATIONS**: Number of generations in one run, used only if SEARCH ALGORITHM is 1.

Note: Increasing GA POPULATION SIZE and GA GENERATIONS will increase the computation time.

**GRID NLOOP**: Number of loops in converging grid search for spectral parameters, used only if SEARCH ALGORITHM is 2. Resolution increases with every loop.

**NGRID FREQUENCY**: Number of grid points in search for corner frequency, used only if SEARCH ALGORITHM is 2.

**NGRID SPECTRAL AMP**: Number of grid points in search for spectral amplitude, used only if SEARCH ALGORITHM is 2.

**NORM**: Norm for computation of residuals in spectral fitting can be set, however, tests show that 1 or 2 produce the same result, and generally default of 1 can be used.

**SEARCH ALGORITHM**: Defines whether genetic algorithm (1) or converging grid search (2) should be used. Converging grid search is recommended.

**SELECT PHASE**: Defines, which phase to use for spectral analysis, choices are: 0 for P by AUTOSIG, 1 for computed P arrival for given location, 2 for computed S arrival, 3 for P from s-file, 4 for S from s-file or 5 for S or P from s-file.

**SEPCTRUM F LOW**: Lower limit of frequency band to be used.

**SPECDURATION CHOICE**: The time window for computation of the spectrum can be given either as a time window starting from the phase onset (0.) or can be defined by a group velocity window (1.).

**SPECTRUM P LENGTH**: Duration in seconds of signal starting from P arrival.

**SPECTRUM S LENGTH**: Duration in seconds of signal starting from S arrival.

**SPECTRUM PRE LENGTH**: Duration in seconds of signal to be included prior to phase arrival.

**GROUP VEL WINDOW P**: Range of group velocities defining time window to be used for P spectrum. Time window is given by (distance/group velocity)

**GROUP VEL WINDOW S**: Range of group velocities defining time window to be used for S spectrum.
21.3. AUTOSIG

Time window is given by (distance/group velocity)

**STALTA NREC/REC**: There are two STA/LTA algorithms, recursive (0.) and non-recursive (1.).

**STATION LINE**: One line with processing parameters for phase detection is given for each channel. The parameters are (also see example below):

- **STAT** - station name
- **COMP** - component name
- **STA** - duration of STA
- **LTA** - duration of LTA
- **RATIO** - trigger ratio
- **MINCOD** - minimum coda required for trigger
- **DTRLE** - de-trigger level
- **FILL** - bandpass filter low cut
- **FILH** - bandpass filter high cut

Example of the parameter file `autosig.par`:

```plaintext
# KEYWORD............Comments.............Par 1......Par 2
#
# spectral parameters
#
SPECTRAL S-Q0  Q0 440.0
SPECTRAL P-Q0  Q0 85.0
SPECTRAL S-QALPHA Q = Q0**Qalpha 0.70
SPECTRAL P-QALPHA Q = Q0**Qalpha 0.70
SPECTRAL KAPPA 0.02
SPECTRAL P-VELOCITY P velocity 6.2
SPECTRAL S-VELOCITY S velocity 3.6
SPECTRAL DENSITY Density 2.8
#
# auto signal processing parameters
#
REMOVE MEAN 1. for true 0.
REMOVE TREND 1. for true 1.
CHAR FUNCT 1. for true 1.
K IN CHAR FUNCT K IN X=Y^2+K*Y'^2. 3.
STALTA NREC/REC rec 0./ non-rec 1. 1.
STALTA SQUARE/ABS square 1. 1.
AUTOCODA SQUARE/ABS square 1. 0.
AUTO PHASE 1. for true 0.
only if no phase
AUTO SPECTRUM 1. for true 1.
AUTO AMPLITUDE 1. for true 1.
AUTO LOCATE 1. for true 0.
NORM 1.
```
SEARCH ALGORITHM 1.=GA 2.=GRID 2.

# window selection
#
SPECTRUM P LENGTH in seconds 5.
SPECTRUM S LENGTH in seconds 2.
SPECTRUM PRE LENGTH in seconds .5
GROUP VEL WINDOW P 5.0 6.5
GROUP VEL WINDOW S 2.7 3.7
SPECDURATION CHOICE 0:SPEC. P/S LEN. 1.
1:GROUP VEL W. P/S

#
# select phase
#
SELECT PHASE 0:auto P 2.
1:synth P
2:synth S
3:P from file only
4:S from file only
5:S or P from file

--- for 3-5, if no phase and AUTO PHASE is 1. use auto phase pick ---
DIST FREQ SELECT 1. 10.

#
# parameters used in the genetic algorithm searching for spectral parameters
#
GA POPULATION SIZE 50.
GA GENERATIONS 250.

#
# grid dimension in case of grid search
#
GRID NLOOP 5.
NGRID FREQUENCY 100.
NGRID SPECTRAL AMP 100.

#
# low filter limit to use for auto spectrum
SECPCTRUM F LOW .05

#
# parameters controlling output
#
CREATE WAVEOUT 1.
SPEC OVERWRITE 0.
PHASE OVERWRITE 0.
# station parameters
#
# STAT- COMP -sta-- -lta-- -ratio mincod -dtrle fill filh
STATION MOL S Z 3.0 20.0 10.0 2.5 1.5 5.0 10.0
STATION MOL B Z 3.0 20.0 10.0 2.5 1.5 5.0 10.0
STATION HYA S Z 3.0 20.0 10.0 2.5 1.5 5.0 10.0
STATION LSA L Z 3.0 20.0 10.0 10.0 9999.0 00.1 10.1
STATION CHTO L Z 3.0 20.0 10.0 10.0 9999.0 00.1 10.1
STATION XAN L Z 3.0 20.0 10.0 10.0 9999.0 00.1 10.1

21.4 AUTOMAG

Program to make automatic readings of WA amplitudes and make spectral fitting for channels for one or several events in an S-file. This is a simplified version of AUTOSIG and the program does not require any parameter file. The program will, by default, process Z channels for which the stations has a P or S reading and the window following the S-phase will be used. Optionally P can be used for spectra and N and E channels can also be used. A check is made for P-window length so if the window will include the S-waves, no spectrum is made. If only a P-phase available, the S-phase time will be calculated from the P-phase arrival time. If the epicentral distance is more than 200 km, the S-phase is assumed to be the Lg phase and an Lg velocity of 3.5 km is used. However if the depth is more then 50 km, S is assumed. In any case, if an S is present, the observed S-time is used.

The program can be used stand alone or in EEV with command automag(am) or command ami (input of parameters).

A special option is to do a grid search for the best Q-parmeters that fit the Brune spectrum using one or several events.

If the S-file has distances, both Ml and Mw can be calculated. The parameter for Ml is taken from STATION0.HYP in local directory or in DAT. If not found, the standard Hutton and Boore(1987) values are used. The additional filter limits for Ml (if any) are taken from MULPLT.DEF, in local directory or DAT. If not filters in MULPLT.DEF, no filters are set. If a filter is used, the amplitudes read are corrected for the gain of the filter at the determined period.

The average Ml and Mw are shown on screen after one run. In order to get the average Ml and Mw in the S-file after using AUTOMAG, an update must be made. However each value is printed out and can also be shown with program PLOTSPEC (see later).

The program has a lot of output on screen when running. This can mostly be used to check why a particular spectrum or amplitude operation failed.

Program input

Interactive input: Give s-file name, spec window and wa window, if zero window length, the corresponding operation is not done, give frequency range for spectral fit, give channel names. If station is called ALL, all Z-channels from all stations with readings (P or S) are used. Alternativey other components can be used for all stations. If not using option ALL, the user must give each channel in which case any channel can be used. Give if P or S-spectrum.

If the spectral window is given, data outside this window will not be used. However, the signal to noise ratio is still check so the actual spectral range might be smaller than the specified range.
CHAPTER 21. AUTOMATIC ROUTINES IN SEISAN

Input from prompt, options are:
S-file name: If only S-file name, all stations are used and default values of window lengths are applied.
s xxx : where xxx is spectral window
w xxx : where xxx is wa window
p : P-spectrum (S is default)
n : use north-south component
e : use east-west component
overwrite : If given, input file will be overwritten, but automag.out also made. Option can only be used with one event. Used with EEV.
grid : Grid search, see below

Examples, input file is sfile:
Automag sfile s 10 : Makes S-spectra only, with 10 s window length
Automag sfile s 10 p : Makes P-spectra only, with 10 s window length
Automag sfile w 30 : Makes wa operation only with window length of 50
Automag sfile s 10 w 30 : Do both
Automag sfile s 10 w 30 n : Do both but use NS channel
Automag sfile s 10 w 30 overwrite : Do both and overwrite sfile
Automag grid : Do grid search, other options cannot be used, see below

Channels to process: If interactive option, each channel must be given or the ALL option used. If non interactive: Stations with P and/or S readings are selected if station has a reading on one of the channels.

Before running AUTOMAG, it might be an advantage to remove all old amplitude and spectral values from the input file. This can be done with program DELS or command DELS in EEV.

Location of waveform data: AUTOMAG will look for waveform files in the standard SEISAN way: Local directory, WAV, default data base under WAV (file names must be standard SEISAN) or in any other directory specified in SEISAN.DEF.

Program flow:
Read RESET TEST value for Ml in STATION0.HYP in local directory or DAT
Read filter values for Ml
Read S-file with readings and location
Get waveform files from S-file
Enter or find station and component for instrument correction
Find S/Lg-time or P-time from readings
Select out a time window from waveform channel starting a bit before the P or S-time (see below)
Find channel number in waveform file(s) corresponding to desired channel
Read the P/S-time window from waveform channel, default 50 s, starting 2 s before P/S
21.4. **AUTOMAG**

Read response file
Prepare response removal, different filters and poles and zeros used for Wood-Anderson response
Correct for instrument response
Automatically pick maximum amplitude and corresponding period
Correct for additional filter if chosen
Write out results in internal s-file, overwrite old results
Do the automatic spectral fitting using a time window of 20 s starting 2 s before the P/S. Grid search is used
Write results in internal S-file, overwrite old results
Write results in automag.out
Write channels processed in automag.list, used for plotting with plotspec

**Grid search**

This option allows the spectral fitting to do a grid search for the best fitting Q₀, Qalpha and kappa. The user will be asked for input file name, grid parameters, spectral window, frequency range and stations. For each group of events in input file, the average residual will be calculated and the results output in file automag_grid.out in order of increasing average residual, see example below. The parameter used in one run is output in automag.par, see example later. The fit can be made with many traces from many events; up to 4000 spectra can be used for each attenuation combination in the grid search. Both P and S-waves can be used. In order to not include very bad results, the results from spectra resulting in a stress drop not in the range 1-150 bar, will not be included in the average. This means that the only data where stress drop can be calculated will be used and the events should therefore be updated before using automag.

Q₀, qalpha and kappa will play off each other, therefore a comparable fit might be obtained with a different combinations and it might not be easy find the one to use. Low Q₀ and high qalpha as well as high Q₀ and low qalpha might be combinations giving similar fit. However, the lower Q₀ will generally give a higher spectral level and therefore a higher magnitude. Similarly, kappa also plays off with Q₀, but to a lesser degree. Hence, for any given Q₀ and qalpha, there is generally a best kappa in the range 0.01 to 0.05. In order to limit number of parameters, an initial value of kappa= 0.02 can be used. For the qalpha parameter, the best fits are usually obtained for qalpha in the range 0.3 to 0.9. For Q₀, the values with a good fit are often in the range 100-400, depending on qalpha. A good fit to the Brune spectrum does not automatically indicate correct attenuation parameters, since the spectrum could have another shape than the theoretical shape, but at least it gives an indication of possible values. Equency range can in most cases be default. However, sometimes the antialias filter cut off the signal in the high end (not corrected by SEISAN) which will give a bad fit so a high cut should be selected. Check first with MULPLT how it looks or run for one event and plot all spectra with PLOTSPEC.

The frequency below 1 Hz at which Q becomes constant cannot be searched for and is a fixed parameter set in SEISAN.DEF. The default value is 1.0 Hz.

There is thus many combinations that give a similar residual and it might be difficult to chose. One option is to average all Q-results with the lowest residual. This can be done with the AVQ program. The program will read all Q-results from automag_grid.out and average the results over a given user specified number of results. This process is then done for all the results in a running average so it is possible to get an idea of how the parameters change with increasing residual. For details on how AVQ calculates the average, see the AVQ program under section CODAQ. An example of running the program is shown
below. It is important that curves that are averaged using the same kappa since the averaging does not take kappa into account. AVQ will also make a plot showing the Q-relations for the Q-curves averaged (the first average) as well as the average curve.

Example of running the AUTOMAG program:

automag grid
Give input S-file
dels.out
Give spec window, enter for def (20)
25
Give spectral range, enter for 0.05-srate/2.5
give max distance, enter for all
Start q0, step Q0 and number of steps
50 50 10
Start qalpha, step qalpha and number of steps
0.3 0.1 9
Start kappa, step kappa and number of steps
0.03 0.0 1
Number of tests is: 90
Continue(n/y=enter)
Station code and component (e.g. BER BHZ), end with blank
Station code and component (e.g. BER BHZ), end with blank Giving station code ALL : Use all Z
channels with readings Giving station code ALL C : Use all C channels with readings
ALL
P or S-spectrum=default (enter)
Output file for the above input:

\begin{verbatim}
q0= 50.0 qa= 0.500 ka= 0.030 nf= 5 re= 0.081 mw= 3.46 f0= 4.19 st= 42.0
q0= 50.0 qa= 0.600 ka= 0.030 nf= 8 re= 0.092 mw= 3.41 f0= 4.18 st= 31.8
q0= 100.0 qa= 0.900 ka= 0.030 nf= 50 re= 0.095 mw= 3.24 f0= 5.11 st= 26.4
q0= 150.0 qa= 0.900 ka= 0.030 nf= 49 re= 0.095 mw= 3.17 f0= 4.01 st= 9.3
q0= 100.0 qa= 1.000 ka= 0.030 nf= 51 re= 0.096 mw= 3.25 f0= 4.18 st= 13.1
q0= 50.0 qa= 0.300 ka= 0.030 nf= 4 re= 0.096 mw= 3.48 f0= 5.80 st= 45.3
q0= 200.0 qa= 0.800 ka= 0.030 nf= 47 re= 0.097 mw= 3.11 f0= 4.23 st= 9.0
q0= 300.0 qa= 1.100 ka= 0.030 nf= 33 re= 0.097 mw= 3.17 f0= 3.42 st= 5.4
q0= 400.0 qa= 1.000 ka= 0.030 nf= 33 re= 0.097 mw= 3.15 f0= 3.51 st= 5.4
q0= 500.0 qa= 0.900 ka= 0.030 nf= 33 re= 0.097 mw= 3.14 f0= 3.58 st= 5.5
q0= 150.0 qa= 0.800 ka= 0.030 nf= 49 re= 0.097 mw= 3.16 f0= 4.90 st= 16.4
q0= 100.0 qa= 0.800 ka= 0.030 nf= 43 re= 0.097 mw= 3.22 f0= 5.94 st= 34.9
q0= 150.0 qa= 0.600 ka= 0.030 nf= 43 re= 0.097 mw= 3.20 f0= 5.82 st= 35.6
\end{verbatim}
nf is the number of fits for each test, mw is average Mw, f0 is average corner frequency and st the average stress drop.

How to find the best Q-parameters, use of program AVQ

avq
File name, enter for automag_grid.out
Min number of fits to use in average, enter for 1

q0= 50.0 qa= 0.500 ka= 0.030 nf= 5 re= 0.081 mw= 3.46 f0= 4.19 st= 42.0
q0= 50.0 qa= 0.600 ka= 0.030 nf= 8 re= 0.092 mw= 3.41 f0= 4.18 st= 31.8
q0= 100.0 qa= 0.900 ka= 0.030 nf= 50 re= 0.095 mw= 3.24 f0= 5.11 st= 26.4
q0= 150.0 qa= 0.900 ka= 0.030 nf= 49 re= 0.095 mw= 3.17 f0= 4.01 st= 9.3
q0= 100.0 qa= 1.000 ka= 0.030 nf= 51 re= 0.096 mw= 3.25 f0= 4.18 st= 13.1
q0= 50.0 qa= 0.300 ka= 0.030 nf= 4 re= 0.096 mw= 3.48 f0= 5.80 st= 45.3
q0= 200.0 qa= 0.800 ka= 0.030 nf= 47 re= 0.097 mw= 3.11 f0= 4.23 st= 9.0
q0= 300.0 qa= 1.100 ka= 0.030 nf= 33 re= 0.097 mw= 3.17 f0= 3.42 st= 5.4
q0= 400.0 qa= 1.000 ka= 0.030 nf= 33 re= 0.097 mw= 3.15 f0= 3.51 st= 5.4
q0= 500.0 qa= 0.900 ka= 0.030 nf= 33 re= 0.097 mw= 3.14 f0= 3.58 st= 5.5
...
Number of curves to average: 90
Running average over how many, enter for average of all?
20

Q0= 155.1 qalpha= 0.83 res= 0.096 mw= 3.22 f0= 4.31 avst= 18.4
Q0= 161.2 qalpha= 0.85 res= 0.097 mw= 3.20 f0= 4.28 avst= 16.6
Q0= 164.2 qalpha= 0.87 res= 0.097 mw= 3.19 f0= 4.26 avst= 15.4
Q0= 176.7 qalpha= 0.86 res= 0.097 mw= 3.18 f0= 4.18 avst= 14.4
Q0= 183.2 qalpha= 0.86 res= 0.097 mw= 3.18 f0= 4.16 avst= 14.2
Q0= 196.3 qalpha= 0.81 res= 0.097 mw= 3.18 f0= 4.22 avst= 14.7
Q0= 202.8 qalpha= 0.84 res= 0.097 mw= 3.16 f0= 4.10 avst= 12.7
...
...
Q0= 243.4 qalpha= 0.55 res= 0.102 mw= 3.13 f0= 4.52 avst= 14.2
Q0= 243.2 qalpha= 0.54 res= 0.102 mw= 3.13 f0= 4.54 avst= 14.2
Q0= 238.8 qalpha= 0.53 res= 0.102 mw= 3.14 f0= 4.50 avst= 14.4
Q0= 241.0 qalpha= 0.54 res= 0.102 mw= 3.13 f0= 4.47 avst= 14.0
Q0= 234.1 qalpha= 0.53 res= 0.103 mw= 3.14 f0= 4.53 avst= 15.0
Q0= 222.7 qalpha= 0.52 res= 0.103 mw= 3.15 f0= 4.52 avst= 15.9
Q0= 216.9 qalpha= 0.53 res= 0.103 mw= 3.16 f0= 4.49 avst= 16.2
Q0= 206.6 qalpha= 0.52 res= 0.104 mw= 3.17 f0= 4.51 avst= 17.3
Q0= 192.4 qalpha= 0.55 res= 0.104 mw= 3.18 f0= 4.56 avst= 19.2
Q0= 182.1 qalpha= 0.56 res= 0.105 mw= 3.18 f0= 4.58 avst= 20.0
Q0= 186.4 qalpha= 0.54 res= 0.106 mw= 3.20 f0= 4.59 avst= 20.8

It is seen that the average parameters, including residual, mw and corner frequency now is a slowly varying function of the average residual. It is also seen that a high Q0 and a high qalpha give a better fit than a low q0 and a low qalpha. If some test have few good fits, they can be eliminated by using the parameter "Min number of fits to use in average". In this example, 20 could have been used.

Example of the output automag.par file

ML constants: 1.000000 1.11 Component used: Z
Spectrum type: S
Spec dur. wa. window: 20.0 50.0
Filter: 0.05 0.00
Max distance: 10000.0
Start Q0, delQ, nQ 100.0 50.0 15
Start Qa, delQa, nQa 0.10 0.10 11
Start ka, delka, nka 0.030 0.000 1
Hardwired parameters set in start of program
Deafult WA window: 50 s
Start 2 s before the S-time
Initial frequency range for spectral analysis: 0.05 to (sample rate)/2.5
WA period must be less than 5 s for results to be saved
Corner frequency must be larger than 0.05 for data to be used
The residual of the spectral fit must be smaller than 0.3 for spectral fit to be used
The Lg-velocity is 3.5 km/s
When doing grid search, the stress drop must be in the range 1-150 for the event to be used.
Parameters from parameter files
Q, kappa, density and velocity come from the spec model in SEISAN.DEF. If no spec model available, the model is taken from MULPLT.DEF. Notice that the frequency dependence of Q from frequency lower than 1 Hz can only be set with the spectral model parameters in in SEISAN.DEF. See also MULPLT and SEISAN.DFF. if the parameter form MULPLT.DEF are used, Q is assumed constant=q0 below 1 Hz.
For explanation of the output of spectral parameters in the S-file, see MULPLT, spectral section.
The Mi constant are taken form STATION0.HYP
The Mi filter constants are taken from MULPLT.DEF
The spectra for one event can be plotted after running AUTOMAG, see program PLOSPEC.

21.5 PLOTSPEC

The program PLOTSPEC is plotting of spectra generated by AUTOMAG.
The command AM in EEV automatically generates the spectra for the corresponding event and read the maximum WA amplitudes. The automatic obtained parameters, within the error limits defined in AUTOMAG, are then saved in the S-file. In some cases, both the automatic spectral analysis and the automatic amplitude can fail. The spectra most often fail because the automatically determined frequency interval has been wrongly selected. The automatic amplitude can fail if the signal to noise ratio is high so amplitudes are determine in the noise. Manual analysis of course spot these problems at once but for the automatic analysis it is hard to judge if a deviating magnitude is real or caused by wrong analysis. PLOTSPEC can therefore plot all the accepted results for the automatic analysis for one event and it is easy to judge the results (see Figure 21.2). Determinations considered in error can then be deleted on the plot and will, when finishing the program, also be deleted in the S-file. The command in EEV to plot is PS.

Program operation
AUTOMAG generates a series of output files intended for plotting with names like BKS.BH.Z.noise, BKS.BH.Z.obs, BKS.BH.Z.synth. The station and component information for all accepted determinations are stored in the file AUTOMAG.list, see example below showing entries for the first 5 channels in Figure 21.2.
CHAPTER 21. AUTOMATIC ROUTINES IN SEISAN

The first line is the header line from the S-file so it does not reflect the magnitudes calculated automatically until an update has been made. Second line is the station and component. Third line is local magnitude and corresponding amplitude (nm) and period(s). Fourth line the frequency interval used for the spectral fitting and the fifth line the moment magnitude. Values that could not be determined are left blank.

The plot has room for 15 pairs of spectra/WA traces. The fitted spectrum is plotted in red and for the WA trace, the trace between the two extremes used for the amplitude observation, is plotted in red. If either the spectrum or the WA-trace is not used for a particular component, the corresponding part of the plot is left blank. If there are more than 15 component plots, several pages can be plotted.

To delete a value, put the cursor in the corresponding plot square and press d. A diagonal line will appear in plot to indicate that a value is to be deleted. When terminating the program, the user will have to confirm the delete of all. There is no undelete option for single channels.

NOTE. Both spectral information and WA amplitudes can be deleted from the s-file with command DELS.

PLOTSPEC is intended to be used from EEV but it will also work outside EEV by giving command PLOTSPEC. The program will then look for the latest AUTOMAG.list in working directory and plot spectra for the FIRST event ONLY. If AUTOMAG has been used with many events, some of the results for the first event might have been overwritten if the same STAT-COMP is used in several events and will not be plotted. PLOTSPEC has one optional argument which is the S-file to use.
Figure 21.2: Plot using program PLOTSEC. The WA amplitude for OSG and spectral estimate for LOF have been deleted. The automatic fits are plotted in red. This event is found in the TEST database.
21.6 Detection program for continuous data, CONDET

The CONDET program is a detection program works on data that is organized in a SEISAN continuous database or a BUD or SeisComp archive. It performs a detection process similar to real-time processing systems, but of course the data is already there. The program works in two steps, first to run a detector on a single channel, and second to detect events that are on more than a minimum number of stations. Possible applications are processing of data from a temporary deployment (e.g., aftershock monitoring, where continuous were recorded without event detection) and adjustment of detection parameters used in real-time monitoring.

The program has three built-in detection algorithms: 1) standard squared STA/LTA, 2) Carl Johnson’s detector (that is for example used in the Earthworm processing system [http://folkworm.ceri.memphis.edu/ew-doc/](http://folkworm.ceri.memphis.edu/ew-doc/)) and 3) correlation with master event. The program writes out a list of detections (file `condet.out`, which gives station name, component code, trigger time and trigger duration), but also a batch file that can be used to extract the corresponding event data from the continuous data (file `extract.batch`). Note that wavetool by default takes data from all continuous databases listed in `SEISAN.DEF`.

When started without any command line options, the program works on all stations/databases given by the STATION parameter. The output file has detections from all stations, and the extract.batch file has extract commands for all detections. This is all required if only one station is available. For more than one station, it is possible to search for times at which more than a minimum number of stations have triggered. This is done by starting the program with the command line argument `-net`. In this mode, the output file `condet.out` from the first run is used and the file `extract.batch` is overwritten. The extract script can now be used to get data for the network detected events. The script can be sources in Unix, under windows run comman in script or rename script to `extract.bat` and then run it. Condet is intended to run without questions and the parameter file `condet.par` must be in working directory. An example of `condet.par` is in DAT.

The input parameters are given in `condet.par`:

**STATION**: give continuous database name, station and component code. If an archive, the data base name is not used.

```
STATION       LICOC  LICO  HH  Z
STATION       LIGLC  LIGL  HH  Z
```

The **BASE TYP** is SEISAN (blank), Archive (arc)

```
BASE TYP  arc
STATION   LIGLC  LIGL  HH  Z
```

**START DATE** and **STOP DATE**: give time interval, can be larger than data availability

```
START DATE   200802270000
STOP DATE    200803122359
```

**WAVEOUT**: Set to 1. to write out waveform files with the original data and trigger channels.

**EXTRACT DURATION**: Length of extraction window in seconds, used in `extract.batch`

**PRE EVENT TIME**: Time to start extract before detection time in seconds, used in `extract.batch`

**INTERVAL**: Length of data segment read at a time. The default is 60 minutes.
DET ALGORITHM: choices for the detection algorithm are STA for squared STA/LTA, COR for correlation and CAR for Carl Johnson’s detection algorithm

MIN TRIG DURATION: Minimum duration the trigger level needs to be exceeded for
MIN TRIG INTERVAL: Only allow for one detection within this time, given in seconds
FILTER LOW: Low cut for bandpass filter
FILTER HIGH: High cut for bandpass filter

If DET ALGORITHM is STA:
STA LENGTH: Short term duration in seconds
LTA LENGTH: Long term duration in seconds
TRIGGER RATIO: Ratio of STA/LTA required for trigger
DETRIGGER RATIO: Ratio to detrigger
FREEZE LTA: LTA can be frozen at time STA/LTA goes above TRIGGER RATIO, 1.=to freeze

If DET ALGORITHM is CAR, see Earthworm documentation for details:
CARL RATIO
CARL QUIET

If DET ALGORITHM is COR:
CORRELATION MIN: Minimum correlation between waveforms of master event and the data required for a trigger
MASTER WAVEFORM: Name of waveform file that is used as master event, the master event is cross-correlated against the continuous waveform data

Network detection parameters:

NET MIN DET: Minimum of detections required from different stations with time window given by
NET WINDOW SEC: Time window for network detection in seconds. NET MAX DELT SEC: Allows the user to specify the maximum time allowed between arrivals on two consecutive stations (in addition to the overall time window for all arrivals given in the preexisting NET WINDOW SEC NET MIN RATIO: Gives the minimum STA/LTA ratio to keep. This lets one use a low STA/LTA ratio in CONDET and then to explore which STA/LTA level to reject during the much faster "CONDET -NET" phase. EXTRACT DURATION: Duration(sec) of extract.
Chapter 22

Calculating b-value, BVALUE

BVALUE is a program to make b-value plots using a NORDIC input file (also compact). The magnitude is selected among all magnitudes on all header lines, but when the first is found, no more are searched for. So only one value is used for each event. A magnitude agency can be specified for the magnitude selection.

A postscript plot file is generated.

The questions are:

Input file name, select.out or collect.out are defaults
   ! Give filename or return
Which magnitude type, C,L,B,W or S, return for no type
   ! C: coda, L: Ml, B: mb and S: surface wave magnitude, W: Moment mag.
      blank: no magnitude type
Magnitude agency, enter for any

Output:
Number of events selected from file: 91
Duration of catalog in years: 0.502

   ! Output number of events selected and duration of catalog
New input:
Magnitude step 1.0, 0.5, 0.25, 0.2 or 0.1
   ! Magnitude step for summing number of events, MUST be one of the above.
Magnitude range m1,m2 for b value and fixed b-value
   ! Range for calculating b value, and the fixed b-value for which a-value is calculated. The a value is calculated as the average of N(i) + bfix * mag(i), where N(i) is the accumulated number of events at magnitude mag(i) and bfix is the fixed b-value.

Output is now:
   n  m1  m2  maxl a  maxl b  sd  lsq a  lsq b  cor  rms  bfix afix sd
### Chapter 22. Calculating B-value, BValue

<table>
<thead>
<tr>
<th>mag</th>
<th>nmag</th>
<th>cmag</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>1</td>
<td>91</td>
</tr>
<tr>
<td>0.6</td>
<td>1</td>
<td>90</td>
</tr>
<tr>
<td>0.8</td>
<td>3</td>
<td>89</td>
</tr>
<tr>
<td>1.0</td>
<td>6</td>
<td>86</td>
</tr>
<tr>
<td>1.2</td>
<td>6</td>
<td>80</td>
</tr>
<tr>
<td>1.4</td>
<td>7</td>
<td>74</td>
</tr>
<tr>
<td>1.6</td>
<td>4</td>
<td>67</td>
</tr>
<tr>
<td>1.8</td>
<td>11</td>
<td>63</td>
</tr>
<tr>
<td>2.0</td>
<td>8</td>
<td>52</td>
</tr>
<tr>
<td>2.2</td>
<td>11</td>
<td>44</td>
</tr>
<tr>
<td>2.4</td>
<td>6</td>
<td>33</td>
</tr>
<tr>
<td>2.6</td>
<td>7</td>
<td>27</td>
</tr>
<tr>
<td>2.8</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>3.0</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>3.2</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>3.4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n: Number of values used</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1 and m2: Magnitude range used for b value calculation</td>
</tr>
<tr>
<td>maxl a and b: Maximum likelihood a and b</td>
</tr>
<tr>
<td>sd: Standard deviation</td>
</tr>
<tr>
<td>lsq a and b: Least squares a and b (the one plotted)</td>
</tr>
<tr>
<td>cor and rms: Correlation coefficient and rms of above</td>
</tr>
<tr>
<td>bfix: fixed b-value given at input</td>
</tr>
<tr>
<td>afix: a-value for above</td>
</tr>
<tr>
<td>sd: standard deviation for above</td>
</tr>
<tr>
<td>mag: magnitude</td>
</tr>
<tr>
<td>nmag: number of events within mag interval</td>
</tr>
<tr>
<td>cmag: cumulated number of events</td>
</tr>
</tbody>
</table>

#### Questions:

Plot(y/n)

! This will make a plot and a postscript plot file. Note that only the last plot on screen is saved. The plot must be sent manually to the printer.
Figure 22.1: An example of a b-value plot. The bars are number of events and crosses the accumulated number of events.

Which b-value
1: Least squares (default)
2: Fixed bvalue
3: Maximum likelihood
! Choice of b-value to plot

Run again with other parameters (y/n)
! Another step length or range can be selected

The final output is:

Output file in bvalue.out
Last plot in file bvalue.eps

The output file bvalue.out contains the same information in the same format as shown in the example above. The file can be used with other plotting programs to make 'nicer looking' b-value plots. An example is shown in Figure 22.1.
Chapter 23

Fault plane solution

SEISAN includes five programs for estimating the fault plane solution, FOCMEC, FPFFIT, HASH (called hash_seisan),EBEL and PINV. The programs are described below, they can all be called from EEV. Fault plane solutions can be plotted with EPIMAP (new from version 9.0), W_EMAP (Windows only) and FOC (see below). Apart from MULPLT, polarities and amplitude ratios can be read in a more efficient way with the programs PLOTPOLARITY, AUTORATIO and PLOTRATIO. The program GMTNOR also makes output which can be used with GMT. In EEV fault plane solutions can be added manually to an event with commands inputfps or ifp.

23.1 FOCMEC

The program can be used to determine double couple earthquake focal mechanisms using polarities and/or amplitude ratios for both local and global earthquakes. The program also provides an interactive graphical display. The existing solution can be plotted without any station data or location being available, however if existing polarities should be plotted, the event must be locatable in order to calculate angles of incidence. Several solutions can be plotted on the same figure in order to compare solutions.

Changes to Focmec

SEISAN used the 2001 version of Focmec and has now updated to the 2017 version and this has implied some significant changes. Polarity weight can now be used. When using polarity weight, each wrong polarity does not count as one but as the normalized (to 1.0) amplitude. This is done in order to give less weight to polarities near the nodal planes. The number of wrong polarities to use for an accepted solution is therefore a real number and calculated as the sum of the weights. In an extreme case, all polarities could be wrong but the sum of the polarities less than 1.0 but generally, according to Snoke, polarity weight improves the reliability of a solution. Amplitude ratios are calculated as before but counted differently in case of observations near a nodal plane where the calculated ratios could be extreme as indicated by NkD in the output (nominator and denominator both below a given threshold, now hardwired to 0.1 for both P and S, before it was 0.05 and 0.15, respsectively). In the old version of Focmec, these ratios were
simply discarded so. E.g. if 8 out of 10 ratios were indicated by N&D, and one error was accepted, only one of the 2 remaining ratios had to be ok for the solution to be accepted. In the new version, every N&D ratio is counted as an error so it is more difficult to get an accepted solution, however unrealistic ratios are not accepted.

NOTE: The old version of FOCMEC is included. Running from EEV, the command is focmec_old and running from the prompt it is focmec

The SEISAN program FOCMEC provides the interface between the database and the program that determines focal mechanisms, which in SEISAN is the program FOCMEC_EXE. This program is written by Arthur Snoke [Snoke et al., 1984] and distributed as part of the FOCMEC package (http://www.iris.edu/pub/programs/focmec/). FOCMEC_EXE is identical to FOCMEC in Snoke’s package and can be easily upgraded (unless formats are changed). Generally the user will use FOCMEC when working with SEISAN data, however, it is also possible to run the original version (see documentation by Snoke: INF/focmec.pdf). Before FOCMEC_EXE is started the user can optionally change the inputfile focmec.run.

The program works with polarities and amplitude ratios. See the MULPLT section on how to read polarities and amplitudes. Note that since amplitude ratios are used, there is no need to correct for instrument response provided the response is the same for the different components (within 5-10 %).

23.1.1 Use of amplitudes

Amplitude ratios are computed from amplitude readings given in the S-file. While amplitude ratios can provide additional constraint on the solution, they should be used with caution. Ideally, the solution should be well constrained by polarities only, and then amplitude ratios can provide confirmation of a solution or help to select one of several equally good solutions. The principle behind the amplitude ratio method is that the effect of geometrical spreading will cancel out when forming the amplitude ratios of S and P waves (or SV/SH) of the same phase type, e.g. Pg and Sg. This leaves the following corrections to be made on the amplitudes before the ratios are calculated.

• Calculate angle of incidence at the station and correct for the free surface effect.

• For local earthquakes, use the calculated travel time for a particular phase to correct for Q. Different Q for P and S can be used and the frequency used is the frequency of the maximum amplitude phase.

• For distant earthquakes, correct for $t^\ast$. Different $t^\ast$ for P and S can be used. The frequency used is the frequency of the maximum amplitude phase.

Different values can be set in file FOCMEC.DEF, which can be located in DAT or working directory.

If there is no definition of Q in FOCMEC.DEF or the file is not there then the attenuation parameters have default values of:

$$Q = 100 \times f^{1.0} \text{ for P and S-waves}$$

$$t^\ast = 1.0 \text{ for P-waves}$$

$$t^\ast = 4.0 \text{ for S-waves}$$

For amplitudes, a model and vpvs are needed. They are defined in STATION0.HYP.

The observations to be made are:
• Rotate the seismogram (if three component record) to get R and T components.
• Read maximum amplitude P-phase and corresponding period on Z, phase P.
• Read the maximum amplitude S-phase (same type) and corresponding period on Z, phase (SV).
• Read the maximum amplitude S-phase (same type) and corresponding period on H, phase (SH).

The wave type Pg/Sg or Pn/Sn has to be given when the amplitude is read. When reading on uncorrected seismograms, MULPLT will want a confirmation that the user wants to save uncorrected amplitudes, since, normally, all amplitude observations in an S-file are in nm. It is possible to filter the signals provided the same filter is used for P and S. Ideally, the amplitude observation should be made at a frequency below the earthquake corner frequency and consequently also the filter high cut frequency should be below the corner frequency.

It is also possible to read amplitudes on the radial component. However, SV amplitudes and phases change rapidly around the critical angle and the amplitudes can therefore be unreliable (see INF/focmec.pdf for details). So, although SEISAN will use the amplitudes read on the radial component, it is in general not recommended to use them. Assuming reading on only Z and H, the following amplitude ratios are calculated:

• SV/P
• SH/P
• SV/SH

In reality, the data only provides 2 independent ratios so ideally only 2 should be used. Since it is hard to know which 2 are the most reliable, SEISAN uses all.

The most reliable amplitudes are P on Z and SH on T so if enough data is available, try not to use SV. However, if only Z-channels are available, SV on Z has to be used.

Unwanted amplitudes can be weighted out with wight 4 in S-file.

FOCMEC also uses polarities on S-waves but this has not been implemented in SEISAN since S-polarities seldom are seen clearly. However, they can be used by editing the focmec.dat file.

Phase names in SEISAN used for amplitudes for FOCMEC have the names AMPG, AMSG, AMPN and AMSN for direct and first arrival (refracted), respectively. For local earthquakes both PG and PN types can be used while for distant earthquakes only PN types can be used.

23.1.2 Automatic reading of amplitudes

Amplitudes (only g-phases) can be read automatically using program AUTORATIO which is implemented in EEV with command ar. Both time domain and frequency domain amplitudes can be read and they have different phase names from the manually determined amplitude phases names so fault plane solution can be made with one of the kind of amplitudes. For details, see AUTORATIO program.

23.1.3 Polarity selection

Any P-phase (first letter of phase name is P) with a polarity (C or D) is used, like P, Pg, PP etc.
FOCMEC can also use polarities of SV and SH, but this has not been implemented in SEISAN.

Automatic polarity determination

Program AUTOPHASE (command ap in EEV) read phase arrivals and polarities. In general it is not recommended to use automatic polarities unless P-phases are very clear. See program AUTOPHASE.

Checking and modifying polarities

Once polarities and/or P-phases are stored in the S-file, the graphical program PLOTPOLARITY (EEV command pol) can modify, add and remove polarities. See PLOTPOLARITY for more details.

23.1.4 Local earthquakes

Any P-phase can be used like Pn and Pg. When few polarities are available, it is an advantage to use both Pg and Pn since these phases have different angles of incidence. Polarities associated with other phases are not used. There is no check if a P-phase has been duplicated.

Amplitude ratios must be determined from the same wave type for example Pg and Sg and the program will only form amplitude ratios from the same wave types. While in principle it should be possible to use ratios determined from refracted waves, generally ratios determined only from direct waves are used since they are easier to identify and have larger amplitudes than refracted arrivals. Particularly the Sn is difficult to identify. This means that the amplitudes readings most often will be made within what is considered the maximum amplitude in the Pg and Sg wave trains. However, the polarity might be read on the first arrival which can be Pn or another refracted arrival.

Distant earthquakes

Polarities of any P-phase can be used (but not pP since first letter is not P ). Using amplitudes require events with clear P and S phases and usually this means reading on broad band records. The amplitude phase names AMPN/SN are used to indicate first arrivals.

23.1.5 Program operation

The program makes a grid-search and finds how many polarities and amplitude ratios fit each possible solution. All solutions with less than a given number of wrong polarities and/or amplitude ratios within given error limits, are then written out and can be plotted. With a cursor, the user can then select the preferred solution, which can be stored in the input file or the database. The program is intended to work from within EEV (option F), however it can also work independently (see below). The program uses an input file called focmec.inp (automatically generated). This is a Nordic format file. Direct waves have angle > 90 and refracted arrivals angle < 90 degrees. If the angle is > 90, the polarity is plotted at an azimuth+180. If the user wants to use FOCMEC as a freestanding program, the angle of incidence information may have to be put in manually in a standard CAT-file, which is then renamed focmec.inp. This can be done automatically by FOCMEC if a hyp.out and corresponding print.out file is available. FOCMEC can also be used to convert angles, like dip, strike and rake to T and P-axis, simply say 'focmec a', where argument a stands for angles and you will be prompted for input.

NOTE that the number of polarity errors is an integer number if no relative weighting of polarities is used and a real number if relative weighting is used, see intro for FOCMEC above and/or FOCMEC manual in DAT.

When the program runs, all amplitude information and corresponding corrections are listed. First there is a question of which type of amplitudes to be used. In the example below, automatic amplitudes are
Number of polarities: 7
Amplitude types: Manual: 0  Automatic: 10  Spectral: 10
Amplitude to use: Manual(1), Automatic(2), Spectral(3) ? 
2

Q: Local: Qp= 142.0**0.71  Qs= 142.0** 0.7  Global: t*(P)=1.10  t*(S)=4.20

<table>
<thead>
<tr>
<th>STAT</th>
<th>C PH</th>
<th>AMP</th>
<th>PER</th>
<th>TRTIME</th>
<th>QCOR</th>
<th>ANGINC</th>
<th>ANGEMG</th>
<th>Fcor</th>
<th>AZ</th>
<th>DIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOO</td>
<td>Z PG</td>
<td>2883</td>
<td>0.28</td>
<td>13.4</td>
<td>1.5</td>
<td>119</td>
<td>40.15</td>
<td>83</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>FOO</td>
<td>T SG</td>
<td>23712</td>
<td>0.24</td>
<td>23.4</td>
<td>2.2</td>
<td>119</td>
<td>40.20</td>
<td>83</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>SUE</td>
<td>Z PG</td>
<td>1448</td>
<td>0.28</td>
<td>13.8</td>
<td>1.6</td>
<td>118</td>
<td>41.14</td>
<td>129</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>SUE</td>
<td>T SG</td>
<td>24591</td>
<td>0.32</td>
<td>24.0</td>
<td>2.1</td>
<td>118</td>
<td>41.20</td>
<td>129</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td>OSG</td>
<td>Z PG</td>
<td>734</td>
<td>0.29</td>
<td>18.2</td>
<td>1.8</td>
<td>106</td>
<td>46.13</td>
<td>199</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>OSG</td>
<td>T SG</td>
<td>11610</td>
<td>0.25</td>
<td>31.6</td>
<td>2.8</td>
<td>106</td>
<td>46.20</td>
<td>199</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>HYA</td>
<td>Z PG</td>
<td>1090</td>
<td>0.28</td>
<td>21.0</td>
<td>2.0</td>
<td>102</td>
<td>47.13</td>
<td>105</td>
<td>144</td>
<td></td>
</tr>
<tr>
<td>HYA</td>
<td>T SG</td>
<td>12161</td>
<td>0.28</td>
<td>36.6</td>
<td>3.2</td>
<td>102</td>
<td>47.20</td>
<td>105</td>
<td>144</td>
<td></td>
</tr>
<tr>
<td>MOL</td>
<td>Z PG</td>
<td>505</td>
<td>0.24</td>
<td>32.2</td>
<td>2.9</td>
<td>94</td>
<td>48.13</td>
<td>59</td>
<td>238</td>
<td></td>
</tr>
<tr>
<td>MOL</td>
<td>T SG</td>
<td>2866</td>
<td>0.28</td>
<td>56.1</td>
<td>6.0</td>
<td>94</td>
<td>48.20</td>
<td>59</td>
<td>238</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>STAT</th>
<th>Ratio type</th>
<th>T</th>
<th>Amp 1</th>
<th>Amp 2</th>
<th>Fcor</th>
<th>LogRat</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOO</td>
<td>SH(T)/P(Z) H</td>
<td>H</td>
<td>23712</td>
<td>2883</td>
<td>0.7</td>
<td>0.93</td>
</tr>
<tr>
<td>SUE</td>
<td>SH(T)/P(Z) H</td>
<td>H</td>
<td>24591</td>
<td>1448</td>
<td>0.7</td>
<td>1.22</td>
</tr>
<tr>
<td>OSG</td>
<td>SH(T)/P(Z) H</td>
<td>H</td>
<td>11610</td>
<td>734</td>
<td>0.7</td>
<td>1.23</td>
</tr>
<tr>
<td>HYA</td>
<td>SH(T)/P(Z) H</td>
<td>H</td>
<td>12161</td>
<td>1090</td>
<td>0.7</td>
<td>1.08</td>
</tr>
<tr>
<td>MOL</td>
<td>SH(T)/P(Z) H</td>
<td>H</td>
<td>2866</td>
<td>505</td>
<td>0.6</td>
<td>0.87</td>
</tr>
</tbody>
</table>

total obs = 12  gap in az = 220.0  gap in ain = 60.0

The abbreviations are STAT: Station code, C: Component, PH: Phase, AMP: Amplitude in count, PER: Period in sec, TRTIME: Travel time in sec, QCOR: Log Q-correction, ANGINC: Angle of incidence at the source, ANGEMG: Angle of emergence at the station, Fcor: Free surface correction for this amplitude, Az: Azimuth from the event to the station, DIST: Epicentral distance in km., Ratio type (see text), T: indicator of ratio type, Amp1 and Amp2: The two amplitudes (count) in the ratio, Fcor is the free surface correction in the amplitude ratio (to be multiplied with ratio) and LogRat is the logarithm of the corrected amplitude ratio used. Total obs: Total number of ratios and polarities, gap in az: the gap of no observations (amplitude and polarity), should be less then 180 deg, gap in ain (amplitude and polarity): gap in angle of incidence, should be as small as possible.

Following, the user get the choices:

Stop  (0)
Plot saved solution(s)  (1)
Plot new solutions  (2)
CHAPTER 23. FAULT PLANE SOLUTION

Plot selected solution (3)
Find new solutions (4)
-1, -2, -3 also plot station

1. This is the solution(s) already stored in the database (S-file). See section "Storing and selecting fault plane solutions" below.

2. Plotting new solution after having used option 4

3. Plotting the selected solution after using option 4
   Using e.g. -1 instead of 1, also plots the stations to help identify them on the plot, see Figure 23.1

4. Starting a search for new solutions
   Option 4 gives the following information and questions:

   There are 7 polarity readings
   Use relative weight, y/n=default
   Maximum number of allowed polarity errors, enter for 1

   Depending on number of data values, 0-5 is a good answer.
   If relative weighting is used, the question about number of polarity errors is:
   Relative weight, number of polarity errors, can be a fractional number, enter for 1.0
   Relative weighting will give more solutions since the numbers summed are smaller than 1.0. So for a good solution, 0.5 might be a good number to use.
   Equivalent for ratios to 'Maximum number of polarity errors', however, error is defined by amplitude ratio error. Number of errors depends on number of observations. For 5 observations 0-1 errors is reasonable.

   Maximum amplitude ratio error, enter for default of .2

   Give maximum allowed difference between observed and computed log amplitude ratio, default is 0.2, which often is a good value.

   Degree increment in search, enter for default 2

   The program will now start the searching and write out on the screen (and in a file) the solutions which fit the requirement of number of misfits. The maximum number of solutions is limited to 100 as a default, or to the value defined by 'FOCMEC MAXSOL' in SEISAN.DEF. At the end, the number of acceptable solutions is written out as well as the minimum number of bad fits. This can then be used for the next search. Now option 0 to 4 can be used again.

   When plotting the solution with option 2, the cursor comes up. Also, the solutions will be printed in text form to the screen, see Figure 23.1

   The abbreviations are Pol: Number of polarity errors for P note it is a real number, SV(not used) and SH(not used), AccR/TotR: Number of accepted to total number of ratio errors, RMS RErr: The RMS error for the ratios used, AbsMaxDiff: The maximum error. See also FOCMEC manual.

   The polarities and amplitude ratios can be plotted on the focal sphere using the same convention as the original FOCMEC program, which is:
23.1. FOCMEC

o = compression
Δ = dilatation
V = amplitude ratio SV/P
S = amplitude ratio SV/SH
H = amplitude ratio SH/P

The user can select a preferred solution by moving the cursor near one of the letters T or P (T and P axis). By pressing T, the program will find the nearest T axis (same for P and nearest P-axis) and corresponding fault plane solution, which can be stored in the database and/or plotted with option 3. If no solution is to be selected, press q for quit. If a solution has been selected, the user will be asked if it is to be saved or not after selecting option 0. The saved solution goes into the focmec.out and from there into the S-file (type F-line) in the database if FOCMEC is operated from EEV and the solution will also be written to fps.out.

When working from EEV, the event will always be located before the FOCMEC program starts up. In the Nordic format the solution is stored simply as strike, dip, rake and number of bad polarities (3f10.1,I5). Aki and Richards convention is used. In addition, the name FOCMEC will be written near the end of the line to indicate that the fault plane solution was made by FOCMEC. The line type is F.

The following files are created:

focmec.dat: Input parameters to FOCMEC_EXE.
focmec.log: Log of the FOCMEC_EXE run.
focmec.lst: More details on solutions
focmec.out: Gives input parameters and solutions
focmec.eps: A Postscript plot file of LATEST plot
focmec.run: Run parameters for FOCMEC_EXE, you can re-run FOCMEC by ‘focmec.exe focmec.run’

23.1.6 Making synthetic amplitudes and polarities to test FOCMEC and HASH

A synthetic input can be generated for FOCMEC. The purpose is, for a given fault plane solution with a given set of stations, to test how well ‘correct amplitudes and polarities’ can be inverted to the known fault plane solution. The procedure is: 1: Select an event with a set of observations and a fault plane solution.

2: Write SYNTET in the F-line in S-file where method is given, using e.g. FOCMEC, this is FOCMEC, replace with SYNTET. 3: Run FOCMEC from EEV. The synthetic polarities and amplitudes will then be generated for the same observations as given in the S-file. These observations are normally written out to the file focmec.dat as polarities and corrected amplitudes. The focmec.dat is then input to the original FOCMEC. The synthetic observations are now made by correcting the focmec.dat file to the theoretical polarities and amplitude ratios corresponding to the SYNTET fault plane solutions. The theoretical values are therefore at the source and not affected by Q and free surface. The original observations in the S-file are not changed. 4: Do the inversion in the normal way.

Since the theoretical values are generated by FOCMEC routines, FOCMEC should exactly return the SYNTET solution. However depending on the number and types of observations, this might not be the case. Before making the solution with FOCMEC, it is possible to edit the focmec.dat file to introduce errors or remove observations in order to test the sensitivity of the solution to errors in input or insufficient input.

All prompt line options for FOCMEC:
a: Calculate angles
o: Plot saved solution, no questions
p: Just run foc preparation
d: Use auto amp as default, use velocity
pd: Options p and d above
s: Calculate theoretical amplitudes from fps
None: Fault plane solution after using HYP

Theoretical amplitudes: Used to calculate amplitudes on the focal sphere for a given fault plane solution, example:

c:\Seismo\PRO>focmec s

strike,dip,rake
30 40 50
Enter az and ain, enter to stop
22 33
p-amp= 0.0301 sv-amp= 0.6091 sh-amp= -0.3259
log ratios, sv/p, sh/p and sh/sv 2.0272 1.7556 0.2716
P-polarity is: C

Enter az and ain, enter to stop

For this option, vpvs is hardwired to 1.74. There is no Q-correction or free surface correction, it is just the amplitude on the focal sphere.

Computer limitations: Total number of polarities must be less than the dimension of array DATA (parameter max_data) for Nordic data (see file seidim.inc in INC directory).

Figure 23.1 shows an example of a fault plane solution calculated with FOCMEC.

### 23.2 FPFIT

This well known program, written by Reasenberg and Oppenheimer [1985], uses polarities to find one or several fps’s (see manual fpfit.pdf in INF). Quoting the manual "Program FPFIT finds the double couple fault plane solution (source model) that best fits a given set of observed first motion polarities for an earthquake. The inversion is accomplished through a two stage grid search procedure that finds the source model minimizing a normalized, weighted sum of first motion polarity discrepancies”. The weighted sum is expressed through the F-factor (0-1) given as output in S-file. A value below 0.5 is a good fit and a value of 1.0 is means a perfect misfit. A station distribution ratio STDR is calculated. Quoting the manual "The station distribution ratio is 0.0 < STDR < 1.0. This quantity is sensitive to the distribution of the data on the focal sphere, relative to the radiation pattern. When this ratio has a low value (say, STDR < 0.5), then a relatively large number of the data lie near nodal planes in the solution. Such a solution is less robust than one for which STDR > 0.5, and, consequently, should be scrutinized closely and possibly rejected”. This value is also written to the S-file. One advantage with
23.2. FPFIT

Figure 23.1: Top: An example of a fault plane solution plot. Symbols are explained in the text. Bottom: A fault plane solution also showing the stations with corresponding polarities.
FPFIT compared to FOCMEC is that formal errors are estimated and usually only one solution is given. The software is found at [http://earthquake.usgs.gov/research/software/](http://earthquake.usgs.gov/research/software/).

The original program FPFIT is left unchanged except for a minor gfortran adaption. FPFIT is an interactive program with many options for parameters stored in a parameter file and different data input formats can be used. In the SEISAN implementation, this has been simplified and a SEISAN driver program FPFIT\_SEISAN is used. This program converts the observations to an input file in hypo71 format, fpfit.dat, makes a parameter file with preset parameters, fpfit.inp and a run file fpfit.run to run the program. After running FPFIT\_SEISAN (either free standing or through EEV with command fp), it is possible to run the original program directly with command fpfit and test different FPFIT parameters, using fpfit.inp as a starting parameter file (default). It is then possible to interactively get information about the different parameters. The hardwired parameters essentially use default settings, ensure the use of all data (e.g. no magnitude-distance restrictions) with the same weigh on all data. In addition, the following is set:

- Search in as fine a grid as the program allows, one deg for fine search.
- Search for multiple solutions, not just the best. Gives an idea of uncertainty.
- Minimum number of polarities to attempt a solution is 6.

Run the program: In EEV, use command fp, first solution is written to S-file. The previous solution of FPFIT will be overwritten. FPFIT in SEISAN implementation can work with both global and local data, while the original FPFIT only works with local data. Outside EEV. See section on composite fault plane solution.

<table>
<thead>
<tr>
<th>Output files:</th>
</tr>
</thead>
<tbody>
<tr>
<td>fpfit.out</td>
</tr>
<tr>
<td>fpfit.fps</td>
</tr>
<tr>
<td>fpfit.pol</td>
</tr>
<tr>
<td>fps.out</td>
</tr>
</tbody>
</table>

Note: There is no check if polarities are read on Z-channel but it is required that the phase is P.

## 23.3 HASH

**NOTE:** internally in SEISAN, the program is called HASH\_SEISAN to avoid conflict with bash command hash.

This program [Hardebeck and Shearer 2002, 2003](#) determines fault plane solutions using P-polarities and amplitude ratios as input, just like the FOCMEC program. The P-amplitude used by the program
is the theoretical radial amplitude and that was obtained by reading \cite{Hardebeck2003}. 
\[ A_P = \sqrt{A_r^2 + A_z^2} \]
and the S-amplitude \( A_S = \sqrt{A_{sv}^2 + A_{sh}^2} \) where \( A \) is amplitude, \( r \) is radial (on rotated seismogram) and \( z \) is vertical. The \( S \) used was the maximum \( S \) on any of the rotated components or \( Z \). This was then compared to the theoretical amplitude \( A_S = \sqrt{A_{sv}^2 + A_{sh}^2} \) where \( sv \) is SV, and \( sh \) is SH. This approximation apparently worked well. The free surface and attenuation correction is not built in, but was replaced by a fixed factor per station, which had to be determined independently. In order to simplify the input, the free surface and attenuation corrected amplitude ratios from FOCMEC are used as input for HASH. The program was modified to use only SH and by using the free surface corrected P on the Z-component, the true P-amplitude is used. Thus only one amplitude ratio is used for each station (SH to P). HASH returns solutions with less than a given number of polarity errors and average amplitude errors less than a given limit. If no solutions are found, error limits are increased and normally many solutions are returned. Using this, an estimate of the best solution is made and likely errors calculated. The advantage with HASH is that it finds one or a few best solutions, while for FOCMEC the user must select one among many. Also HASH will not completely change the solution by one wrong amplitude ratio, since the average of the amplitude errors is used as selection criteria and not a single amplitude. FOCMEC does not give any estimate of the errors in the solution. HASH calculates an estimated error; however that requires an input where each event has been located with e.g. 10 different likely input models and all data is used as input in order to get estimate of fault plane solution uncertainties generated from the model. This was not done in the SEISAN implementation so only the error estimated from the spread in solutions is used. This might lead to smaller error estimates as compared to the original HASH implementation. The SEISAN HASH implementation is a simplified implementation compared to the original HASH with many parameters hardwired, see hash.seisan.for for implementation details and changes. Like FPFIT, the F-fit function is calculated (called weighted fraction of polarity misfits) and similarly the station distribution ratio (see FPFIT). Both values are given in S-file as well as the average amplitude error. For more information, see the HASH manual hash.pdf and FPFIT manual fpfit.pdf in INF. The software is found at [http://earthquake.usgs.gov/research/software/index.php](http://earthquake.usgs.gov/research/software/index.php) HASH does not estimate errors in strike, dip and rake but errors in fault plane and auxiliary plane (degrees). HASH, like FOCMEC uses the Vp Vs velocity ratio to calculate amplitudes. In versions of SEISAN before 10.6, this was hardwired to 1.74. Now it is read from the input file focmec.dat also used with FOCMEC. This ratio is in turn read from STATION0.HYP.

### 23.3.1 Running HASH from EEV

Polarities and amplitudes are picked like for FOCMEC. When running the program, the amplitudes are corrected like for FOCMEC (actually done by FOCMEC) so the Q-correction will use the Q-relation given in focmec.def (see FOCMEC description above). The same output, as for FOCMEC, with the available amplitudes, their ratios and corrections will be shown and the control is the handed over to HASH. The questions are:

- **Grid angle for focal mech. search, enter for def 2**  
  Comment: Smallest is 2
- **Max number of polarity errors, default is 0**  
  Comment: Default
- **Max average error in amp rat, log10, def 0.1**  
  Comment: default 0.1 selected
- **Enter angle for computing mechanisms probability, def is 60**  
  Comment: Default 60 deg. Sel.
- **Enter probability threshold for multiples, def is 0.1**  
  Comment: Default 0.1 selected
Now is following the FOCMEC amplitude information, not shown ...

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of polarities is</td>
<td>11</td>
</tr>
<tr>
<td>Number of amplitude ratios is</td>
<td>5</td>
</tr>
<tr>
<td>Minimum number of polarity misfits overall</td>
<td>0</td>
</tr>
<tr>
<td>Minimum average amplitude error overall</td>
<td>0.13</td>
</tr>
<tr>
<td>New number of pol. misfits inc. extra</td>
<td>1</td>
</tr>
<tr>
<td>New average amp limit inc. extra</td>
<td>0.23</td>
</tr>
<tr>
<td>Minimum average amplitude error for pol ok</td>
<td>0.27</td>
</tr>
<tr>
<td>New average amp limit is</td>
<td>0.37</td>
</tr>
<tr>
<td>Number of solutions found</td>
<td>92</td>
</tr>
<tr>
<td>Strike, dip, rake</td>
<td>197.3 66.5 -157.4</td>
</tr>
<tr>
<td>Fault+aux plane uncertainty</td>
<td>23.2 10.3</td>
</tr>
</tbody>
</table>

Explanations of input:
The "mechanism probability" is the probability that the real mechanism is "close" to the preferred mechanism, within "angle for computing mechanisms probability" where angle define "close." If there are clustered outliers, alternative solutions (or "multiples") are found based on those outliers. You can set the minimum probability for the multiples (i.e. ignore multiples with a low probability.)

23.3.2 Explanations of output

Minimum number of polarity misfits overall: Minimum number of wrong polarities for anyone of the grid points disregarding amplitude fit. This is the number of polarity errors to find a solution without amplitudes.

Minimum average amplitude error overall: The minimum average log error for any grid point disregarding errors in polarity.

New number of pol. misfits inc. extra is: The new limit for polarity errors.

New average amp limit inc. extra: Based on the above, a new amplitude ratio error limit is set.

Minimum average amplitude error for pol ok: The new error limit considering polarities within limit.

New average amp limit is: In order to get sufficient solutions, the amplitude error limit is increased to this value.

Output files: Hash.seisan.out: A summary of the solutions(s).
Fps.out: The solution(s) in SEISAN format.

Storing and selecting fault plane solutions: Format errors estimates and quality.
The fault plane solutions are stored in the S-file. Different programs give somewhat different parameters and sometimes the same output field is used for different parameters. Some programs give strike of dip instead of strike of fault plane, but values used in SEISAN are converted to strike of fault plane. Each program is indicated with its own name like "HASH F" at the end of the F-line. If no characters are written in the blank space, any new solution will overwrite the old one. However if anything is written like "HASH 1P", any new solution will create a new line in S-file. This is also the case if a quality indicator is written (see below). An example is:
In this example, there are 4 solutions made by the 4 programs and the solution made by FPFIT has been selected as a prime solution with quality A. The content and format is:

Type F Line (Optional): Fault plane solution

Columns Format Description

1:30 3F10.0 Strike, dip and rake, Aki convention
31:45 4F5.1 Error in strike dip and rake (HASH), error in fault plane and aux. plane (FPFIT)
46:50 F5.1 Fit error: FPFIT and HASH (F-fit)
51:55 F5.1 Station distribution ratio (FPFIT, HASH)
56:60 F5.1 Amplitude ratio fit (HASH, FOCMEC)
61:62 I2 Number of bad polarities (FOCMEC, PINV)
64:65 I2 Number of bad amplitude ratios (FOCMEC)
67:69 A3 Agency code
71:77 A7 Program used
78:79 A1 Quality of solution, A (best), B C or D (worst), added manually
80:80 A1 Blank, can be used by user

Quality indicator: The indicator can be any character, but usually A to F is used with A as the best. It is up to the user to manually assign a quality indicator. Events can later be selected based on quality indicator. Programs SELECT and FOC use quality indicators. The quality indicator as well as the selection of the prime solution can be select by command fq in eev. An example is given below:

Fault plane solutions for this event 1 180.3 55.1 -123.7 5.0 4.4 0.04 0.67 0.38 SJA HASH A 2 6.1000 48.4400 -48.0700 2 SJA FOCMEC 3 191.0 51.0 -110.0 SJA DREGER 4 185.0 48.0 -121.0 HRV CMT A
Give fps number to be prime solution, enter for no change

23.4 PINV

This program makes a preliminary best fault plane solution based on polarities and is intended as a help to other methods of fault plane solution. The original program was written by Suetsugu and some information is found in [Suetsugu, 1998]. A copy of this report, which also gives general information about fault plane solutions, is available as foc.pdf at [http://iisee.kenken.go.jp/lna/?mod=view&cid=S0-250-2007](http://iisee.kenken.go.jp/lna/?mod=view&cid=S0-250-2007)

To run the program from EEV:

Command pi will locate the current event and then start PINV (stands for P-inversion). PINV is hardwired to use hyp.out as input file and it will use all polarities from P-phases (capital P as the first letter). The result of the inversion is written out on the screen and in file pinv.out. The strike, dip and rake and number of wrong polarities is also written to the S-file provided at least 5 polarities are available, however PINV will make an inversion with any number of polarities and write the result to the
screen. In the S-file, the result is written as an OF-line giving the source of the inversion as PINV. A new inversion will overwrite the previous solution. This means that a PINV solution will be additional to the solution given by the F-line and therefore not considered as prime. It is also possible to directly compare the solution to the solution obtained by FOCMEC.

To run program outside EEV.

The program can run with one or many events (composite solution). First locate event(s) with HYP, then give command ‘pinv’ and the inversion is made. All polarities in the hyp.out file are used. The result only goes the screen, hyp.out, fps.out and pinv.out.

Principle of operation:

Moment tensor inversion can be done most simply using amplitudes as observed on the focal sphere. In PINV, polarities are considered to be amplitudes of +1 or -1 corresponding to compression and dilation, respectively. This is a gross oversimplification since there will be large variation of real amplitudes over the focal sphere. This input data of ‘amplitudes’ is then inverted to get the moment tensor under the restriction of finding a single double couple. Despite the simplification, the advantage of this method is that it very quickly gives a ‘best’ approximation to the fault plane solution. This “best” solution, particularly with few data, might be just one of many possible (see FOCMEC), but it serves to give an idea of a possible ‘best’ solution and it is in general consistent with the observations [Suetsugu 1998]. Unfortunately PINV does not give an error estimate to judge how reliable the solution is. It is therefore not recommended to use PINV to obtain prime fault plane solution, but rather as a help to select a solution when using FOCMEC unless much well distributed data is available. In some cases, FOCMEC will find a solution where all polarities fit, while PINV will get a similar solution with some polarity errors. This can be explained by PINV using the assumption of +1 and -1 amplitudes and thus an overall fit to amplitudes near nodal planes might be difficult. The original input to the PINV program has an option to give zero amplitude of observations judged nodal, however in our experience it is hard to judge if a first polarity is nodal or just has a small amplitude due to path (e.g. Pn) so this option has not be included.

An example of a run is:

Number of data used for inversion= 8
Absolute pseudorank tolerance 0.001210 Pseudorank 5
Strike, dip, rake 72.3 38.9 34.6
Consistent data: 8
Inconsistent data: 0

There were 8 observations which all fitted. The pseudorank indicates how many parameters can be determined in the inversion. In this case 5 since there are 8 observations. If less than 5 observations, the pseudorank will be less than 5. The Absolute pseudorank tolerance is a measure of the fit.

### 23.5 Fault plane solution program to use and some advice

The different programs all have advantages and disadvantages. FOCMEC uses the most data since it can use more amplitude ratios than HASH but it might be difficult to find the ‘correct’ solution since a small change in input limits might make a large change in output. If a few of the amplitude ratios are very wrong, an unrealistic high ratio limit must be used and many errors allowed. This problem is avoided by using HASH since the limit is the average amplitude ratio error and not the number of errors. If only
working with polarities, all 4 programs can be used. PINV gives a very quick solution which can be used as an indication of a possible best solution, however for final results one of the other 3 programs should be used. It is often a good idea to compare the results from the different programs. Ideally they should all give the same result, but there will be difference due to different methods used and different data, however if solutions are very different, the solution might not be very stable.

Polarities are often not very clear so in general, when using FOCMEC, one should allow for polarity errors. Even if polarities are very clear, there might be errors since the theoretical ray calculated by HYP does not correspond to the observed ray. This can be due to a wrong model or a wrong hypocentral depth. If e.g. the program calculates that the first arrival is Pg, but the arrival is very emergent, the first arrival might be a refracted arrival due to wrong model or depth. In some cases a clear Pg arrival might be seen a bit later and can be used instead.

In some cases due to incorrect model, the hypocenter must be fixed to a depth different from what it locates to naturally in order to get a solution.

Note that in order to compare FOCMEC and HASH, amplitudes should only be used on Z and T since HASH cannot use amplitude on Z. In any case, these amplitudes are the most reliable. In HASH, the number of solutions will depend mainly on the error limit given for the amplitude ratios. If very few solution are found, the solution might be less reliable so increase the limit then. A few hundred solutions is a good number. The results from two solutions can be compared with EEV command ‘fd’ which calculates the difference it P and T orientations. It is easy to compare the solutions. Run each program in EEV, then plot using command fo. Each solutions will be plotted in a different color, see Figure 23.3. If doing composite solutions, use program FOC with input from fps.out.

23.6 Composite fault plane solution using any of the programs FOCMEC, FPFIT, HASH and PINV

For some small networks, the number of stations is too small to make a fault plane solution using polarities from a single event and even using amplitudes, there might not be enough data. Under the assumption that the underlying stress field will generate events with similar fault plane solutions, a group of events can be used together to make one fault plane solution which will represent an average of the supposedly similar solutions. The advantage of using several events with different azimuths and angles of incidence is that the observations will be well spread out on the focal sphere. The four fault plane solution programs in SEISAN can all be used to make a composite solution and both polarities and amplitudes can be used. The procedure is similar for all programs:

1: Select the events to be used together in one file using e.g. SELECT or the graphical options in EPIMAP.

2: Locate the events with HYP. There are now two output files, hyp.out and print.out available for input to the fault plane solution programs. The fps program must be executed in the directory of the print.out and hyp.out. Once the fault plane solution has been made, the solution is available with the first event in hyp.out.

3: Do the fault plane solution:

   - FOCMEC: Run with command ‘focmec’.
   - FPFIT. Run with command ‘fptfit’.
- HASH: Run with command 'hash'.
- PINV: Run with command 'pinv'.

4: Plot the composite fault plane solution: The solution can be plotted with command 'focmec o'. Only polarities will be plotted. The solution can also be plotted with plotfoc using hyp.out as input. In this case no polarities are shown. The solution with the first event in hyp.out is used. No agency is written for the composite solution. A previous solution without agency and quality is overwritten. If all 4 programs have been used, all 4 solutions will be present in hyp.out and plotted together. It is possible to plot stations corresponding with all polarities with program focmec. Then type 'focmec' and when the menu 0-4 comes up, chose -1.

The solution(s) will also be written to the cat-file fps.out in standard SEISAN format. For each run of a program, the solutions accumulate in fps.out. This can be used to compare solutions from different programs, see FOC. The solution can also be plotted with PLOTFOC. An example of the plot with FOCMEC is seen in Figure 23.1.

Finally the solutions will also be written to hyp.out and can be plotted with PLOTFOC, however in that case only the mechanisms are shown.

NOTE: All fault plane solutions made from the different programs are accumulated in file fps.out. Figure 23.2 shows an example using polarities from 5 events.

23.7 Moment tensor inversion program, INVRAD

The program is written by John Ebel [Ebel and Bonjer, 1990] for moment tensor inversion for very local events. The program uses instrument-corrected amplitudes of the direct (upgoing) phases of P, SV and SH phases and makes a linear inversion for the moment tensor. The program then finds the largest double couple component of the traceless moment tensor. For more details see file invrad.txt in the INF directory.

The original program has been slightly modified in input and output to be integrated with EEV in SEISAN. The steps to get the fault plane solution are:

Select the event from EEV

1. Plot each trace and select preferably the first clear amplitude of the direct wave. Mark the amplitude as usual and associate the amplitude with amplitude phases AMPG or AMSG (direct phases). This will create a separate line with amplitude readings only. The polarity must also be indicated on a separate phase, which must be Pg or Sg since the inversion program uses the polarity of the amplitude. The amplitudes MUST be picked on instrument corrected traces if all instruments do not have the same response function. At least 5 amplitudes must be selected. S phases picked on vertical or radial components will be considered SV while S-amplitudes picked on transverse components will be considered SH. Phases picked on NS or EW component cannot be used. If these new phases are not to be used for location, they can be weighted out.

2. Update event with command update to make distance and azimuths available.

3. Use command INVRAD to do the inversion. This command does several things hidden for the user:

- Creates the model input file for INVRAD called invrad.mod. This file is created from the STATION0.HYP file, either from the current directory or DAT.
Figure 23.2: Composite solutions using 5 events and the 3 programs FOCMEC, FPFIT and HASH. Only polarities are used.
- Creates the data input file for INVRAD called invrad.inp. This file is made from the current database file (S-file) by extracting all amplitudes associated with Pg and Sg amplitudes and converts to P, SV or SH amplitudes in microns. The depth of the event is taken from the S-file header and the estimated error is fixed to 0.1 micron.
- Runs the INVRAD program which produces the invrad.out file
- Reads the invrad.out file to get the fault plane solutions which overwrite the current fault plane solution in the S-file. If you do not want to get the current solution overwritten, put a character in column 79 on the solution, see also focmec program.

The fault plane solution can then be plotted with FOCMEC.

### 23.8 Plotting fault plane solutions

There are many ways of plotting fault plane solutions in SEISAN: Through EEV (a single event), program FOC (many events), program PLOTFOC (single event without station information), program MOPAD which also can plot a moment tensor solution, program EPIMAP (many events), option MAPS in EEV (one event on a map) and W_EMAP (many events). The input file is in all cases a CAT-file. In addition, using program SEIGMT, a file to be used with GMT is prepared, however the use must make his own script. Only through EEV is it also possible to plot the observations.

Data can also be exported in GMT format with programs SEIGMT and MOPAD.

Using EEV
Command fo will plot all events in S-file. This can be a useful ways of comparing solutions obtained by different programs, see Figure 23.3. It is a requirement that the event can be located and polarities will be shown, not amplitudes. Command foo will work as fo but no polarities are shown. There is no requirement for the event to be located so foo can be used if an S-file only has a header line and fault plane solution(s).

Command fh will plot a moment tensor solution with MOPAD program.

Using FOC
See under FOC for how to run the program.

The plot is seen in Figure 23.4.

Using PLOTFOC The program asks for input of

W_EMAP (Windows only) plots the solutions as seen in Figure 23.5. In this case the simplest is to give command w_emap file, where file is the CAT file with fault plane solutions. See W_emap manual in INF. NOTE: Some versions of W_EMAP plots some of the fault plane solutions with inverted color e.g. inverse fault becomes a normal fault).

The EPIMAP plot for the same events is shown in Figure 23.5. See EPIMAP for more explanation. EPIMAP can also plot the fault plane solutions in a section, the solutions are still seen in the horizontal plane.

### 23.9 FOC: Plot many FPS, stress inversion and Rose diagram

The SLICK program
Figure 23.3: Compare fault plane solutions from different programs. For explanation of symbols, see FOCMEC.

Figure 23.4: Example of plotting many solutions. Each solution is given with number, the fault plane solution and the quality (A-E). Up to 24 solutions can be plotted on one page.
This program is part of the Slick package doing the following quoting the author \cite{Michael1984} "The slick package uses fault slip data (either field observations or from focal mechanism) to find the stress tensor that best explains the observations. Inputs are the orientation and slip direction of a set of fault planes. Outputs are the orientation and shape of the stress ellipsoid, including confidence regions, and statistics used to judge the success of the inversion. This method uses the linear inversion algorithm and non-parametric bootstrap statistics". The software is available at \url{http://earthquake.usgs.gov/research/software/index.php}

In SEISAN, only the inversion part has been implemented so the error analysis is missing. Program SLICK can be run as a separate program, but is normally run as part of FOC which prepares the input for SLICK and plots the output. The method is explained in \cite{Michael1984} where also examples are given with data available at the above web site. Running SLICK: slick "file", where "file" is a file with strike of dip, dip and rake. An example input file is:

\begin{verbatim}
Strik dp Dip Rake 
 203.0  51.0  137.0
 280.0  85.0 -161.0
...
\end{verbatim}

Note that in SEISAN, strike of fault plane is used so the strike of the dip is strike of the fault plane+90 degrees. The output is "file.out" which gives the found stress tensor and the fit to the data, for details see \cite{Michael1984}. The stress tensor has a corresponding slip angle, (average slip) and for each event the difference in slip angle for the individual event and the average slip is calculated as well as the average difference and standard deviation. When running SLICK with FOC, an input file foc.slick is made for selected events (making the corrections to strike of slip) and the output file is foc.slick.out. FOC also plots the direction of maximum compressive stress s1, minimum compress stress s3 and null axis s2. In the example below s1 has max value of 0.68 and strike and dip are 19 and 34 respectively. S3 has strike and dip of (113, 5) and s2 (-149, 56) respectively. The average fit angle is 59 with a standard deviation of 51, a bad fit.

\begin{verbatim}
stress tensor is:
-0.290526  0.236582 -0.146602
 0.236582  0.293028 -0.438347
-0.146602  -0.438347  0.00250278
\end{verbatim}
23.9. FOC: PLOT MANY FPS, STRESS INVERSION AND ROSE DIAGRAM

```
eigenvalue vector: E,N,UP,direction,plunge
0.686771 -0.273399 -0.784281 0.556917 19.205739 33.820430
-0.37516 -0.917882 0.385856 0.0927811 112.725966 5.320093
-0.31161 0.287656 0.485818 0.82537 -149.270892 55.589101
variance= 0.283314
phi value= 0.940156

dip direction, dip, rake, fit angle, mag tau
  203.0   51.0   137.0   166.4   0.11
 280.0   85.0  -161.0   167.9   0.19
...
  13.9   68.1  -85.7   11.0   0.46
  14.0   70.0  -130.0   33.3   0.48
fit angle mean= 59.156784 standard deviation= 51.527176
for f=0.8 I= -2.242677 , std. dev.= 1.609795 D norm= 0.248640
avg tau=xx , std. dev.= xx
```

For a complete stress analysis it is recommended to also do the error analysis using the complete slick package or e.g. the program ZMAP (not a SEISAN program, uses MATLAB, found at [http://www.earthquake.ethz.ch/software/zmap/ftp](http://www.earthquake.ethz.ch/software/zmap/ftp)). FOC writes a file which is formatted for input to ZMAP. However doing stress analysis as implemented in SEISAN gives a good impression of the consistency of the fault plane solutions in a particular area. It is recommended that at least 10 events are used for inversion.

FOC program

FOC is a program doing different things with fault plane solutions given in a CAT-file: Converting data to other formats, plotting many solutions, running the SLICK program and displaying the results, plot P and T axis for many events and make statistics of polarities. The input is:

```
foc
Give input file
collect.out
   Quality, ABC.., up to 5 chars, enter for all
   AB Comment: Different qualities can be selected
   Cumulative(c) or individual misfit(def) Comment: See later
   Plot all solutions selected (Y=enter/n) Comment: Analysis can be done without plotting all
   N
```

The plot of the many fault plane solutions is seen in Figure 23.4. After plotting the fault plane solutions, a plot comes up plotting the location of the P and T axis and the results from SLICK, see Figure 23.6.

Output files:
Foc.out: P and T axis for all events, can be used as input to make rose diagrams.
Foc_events.out: The events used based on quality selection
Foc_pol.out: Statistics of polarities:

```
Stat  C   D
AZ05  3   2
```
Figure 23.6: Left: The position of the T-axis given by the event numbers. The triangle is the SLICK minimum compressive stress direction and the circle is the null axis. Right: Corresponding for P-axis and the triangle is now the maximum compressive stress direction. Top: The misfit for each event as a function of event number. This figure can also show the cumulative misfit, see example run of FOC.
For each station the different polarities are counted and a sum of the consistent polarities are given at the end. Foc.zmap: Input file in format used by ZMAP, notice direction of slip is used instead of strike of fault, see SLICK.
Foc.slick and foc.slick.output: See SLICK.
NOTE: FOC uses the first instance of the fault plane solution found in file for a particular event.

23.10 PLOTFOC, plotting fault plane solution without need for station information

The program can plot all fault plane solution in an S-file without any need for location, polarities or any other station information. The plot looks like the plot for FOCMEC. The program can be started without argument and the user is asked for the name of the S-file or CAT-file. The program can also be started with the input file as argument. If the input file contains several events, it is possible to plot one event at a time, one after the other. Optionally, all events can be plotted superimposed. This can be used for comparing the fault plane solution from two different events, superimposed. Plotting fault plane solutions side by side can be done with program FOC. The program can also be started form EEV with command 'foo'. In this case only one event can be plotted.
The plot will list number of polarities and azimuthal gap of the polarities. The azimuthal gap is calculated taking into account the azimuths as plotted on the focal sphere.

23.11 MOPAD, plotting a moment tensor solution and more

A Python program for plotting the focal sphere of seismic sources, both pure double couple and non DC sources described with the moment tensor. The program is made by Krieger and Heimann [2012] and available at [http://www.larskrieger.de/mopad/](http://www.larskrieger.de/mopad/). The program is included with SEISAN and started with command mopad. The program has many options for plotting and manipulating seismic sources using either the moment tensor or the strike, dip rake as input. For examples see [http://www.larskrieger.de/mopad/](http://www.larskrieger.de/mopad/). To see the options type mopad -h. MOPAD is the only program in SEISAN that can plot moment tensor
solutions. It is implemented in EEV (command fm) with the only option of plotting the moment tensor solution as given in the S-file.

See Figure 23.7 for an example. The corresponding S-file is

<table>
<thead>
<tr>
<th>Year</th>
<th>Day</th>
<th>Month</th>
<th>Time</th>
<th>Action</th>
<th>Status</th>
<th>ID</th>
<th>MT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1976</td>
<td>1</td>
<td>1</td>
<td>129</td>
<td>53.4</td>
<td>HRV</td>
<td>7.3HRV</td>
<td>6.2bHRV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17-06-02</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19:06</td>
<td>0P:ff</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ID:19760101012953</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>202.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HRV</td>
<td>30.</td>
</tr>
<tr>
<td>1976</td>
<td>1</td>
<td>1</td>
<td>129</td>
<td>53.4</td>
<td>HRV</td>
<td>7.3HRV</td>
<td>6.2bHRV</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17-06-07</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19:06</td>
<td>0P:ff</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ID:19760101012953</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>202.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HRV</td>
<td>30.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GlobCMT</td>
<td>93.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.956E+20</td>
</tr>
</tbody>
</table>

MT 7.680 0.090 -7.770 1.390 4.520 -3.260 HRV 19 0.956E+20 0.956E+20
Figure 23.7: Example of plotting the focal sphere of a moment tensor solution
Chapter 24

Plotting and changing of polarities, PLOTPOLARITY

PLOTPOLARITY takes the s-file name as input and makes a plot of P-wave onset with polarity readings if available. The program provides an efficient way to plot and change polarities used to determine fault plane solution. Only vertical channels of data are shown. If changes are done, the program will save, if the user wants to, these back to the s-file. The plots can extend over several pages when there are many traces.
PLOTPOLARITY can be started from EEV using command 'pol', or from the command line using either, e.g.:

```
plotpolarity 09-2257-08L.S201508
```
to plot traces which have polarities in s-file already, or with:

```
plotpolarity 09-2257-08L.S201508 all
```
to plot all vertical channel data. This is the default option when starting the program from EEV. The program asks for some input at the startup:

```
total window duration (default is): 2
```
This is the total window duration that will be displayed around the P arrival, which is taken from the s-file.

```
P-onset position as percentage of total duration (10-90; def 50 = middle): 60
```
This number gives the position of the P arrival within the time window as percentage of the total window. Setting this to 0 would mean that the signal starts from the P arrival, 50 means that P will be in the center of the window, and so on. Numbers less than 10 and greater than 90 will be reset to 10 and 90, respectively. One typically wants enough pre-phase time to see how the signal is changing.
window length in seconds for averaging (default is 0)

This allows to average the samples for the given time period starting the given length prior to the time of a sample. The averaging will act as a filter without a phase response, and can be looked at for noisy signals, best only to confirm if the signal seen on original trace is still the same when applying averaging filter.

Next the plot comes up, and the polarities can be inspected and changed, using keys c=compression, d=dilatation and r=remove. In case of a change, a message is written to the terminal, and the graphics are updated. For each trace, the station name is shown together with the phase, the polarity (empty if not available), the distance and azimuth. The graphics are written to a Postscript file polarity.eps, an example is given in Figure 24.1.
Figure 24.1: Sample plot of PLOTPOLARITY.
Chapter 25

Measuring amplitude ratios, AUTORATIO

The program runs on an s-file and measures P and S amplitude in both time and frequency domain. The measurements are done for a number of frequency bands and the results are obtained by averaging. The results can be used with FOCMEC and HASH to determine the fault plane solution. Data windows are selected based on the arrival time picks in the file and have a user selectable length (default 2 sec). The program adds 20% of time as pre-phase time. AUTORATIO can convert to displacement or velocity units and allows for filtering of the data. The user can set a maximum hypocentral distance, default is 100 km. While the program would work with Pg/Sg or Pn/Sn, it is probably wise to work with short distances and direct phases only. The results are written to the autoratio.nor output file in Nordic format. If called from EEV (command ar), the results are written directly to the s-file. Another output file, autoratio.out, lists all the amplitude readings.

The data processing steps are as follow, first for P, then S signal:

- Read the complete trace, for P this is the vertical channel, for S the horizontal channels are read and rotated into radial and transverse using the computed back azimuth
- Taper and remove DC, possibly filter and convert to ground motion
- Select time window
- Read amplitude of time domain data: P on Z, SV on Z and SH on T.
- Apply FFT to compute spectra
- Read spectral amplitude by fitting with Gaussian curve
- Do the above for different frequency bands and average
- Write out the results

The following amplitude names are used in the Nordic output file:

- ATPG: amplitude Pg time domain
• ATSG: amplitude Sg time domain
• ASPG: amplitude Pg spectral domain
• ASSG: amplitude Sg spectral domain

The following amplitude names are used in the Nordic output file:

• StoPt: Sg to Pg ratio time domain
• StoPs: Sg to Pg ratio spectral domain

There are thus 2 S-amplitudes and one P-amplitude and the fault plane solution program will make 3 amplitude ratios, see FOCMEC and HASH. The least reliable S is probably S on Z. The program has a parameter s on Z which can be set to not reading SV on Z. The program must then be recompiled.

NOTE: Each time the program runs, all old automatic amplitudes are deleted. Manual amplitude pics are not deleted.

25.1 Plotting amplitude ratios, PLOTRATIO

This program plots the output from AUTORATIO. It shows the time domain traces and spectra for P and S, respectively, together with the amplitude measures and ratios described in the AURORATIO section. An example is given in Figure 25.1
25.1. PLOTTING AMPLITUDE RATIOS, PLOT_RATIO

Figure 25.1: Sample plot of PLOT_RATIO.
Chapter 26

Waveform inversion

26.1 Moment Tensor inversion in SEISAN

Introduction
The moment tensor inversion for regional earthquakes implemented in SEISAN uses the well tested Dreger code [Dreger, 2003]. The software has been integrated into SEISAN to take advantage of all the parameters already being part of the SEISAN data base like response, hypocenter and station parameters as well as SEISAN’s ability to do instrument correction and filtering. All operations take place through EEV including an optional search for the best hypocentral depth. A tutorial for the original Dreger software including basic information on the principles of the software is found in INF, mt_dreger.pdf.

What data to use
The program works for regional distances (up to 2000−3000 km) where the model can be approximated by flat parallel layers. The inversion will generally work best for events large enough to produce low frequency signals (f<0.1 Hz) which means surface or sometimes S-waves (at short distances or for deep events). However theoretically there is no lower magnitude limit since the source time function is a point source. A simple test to see if the data potentially can be used at low frequencies, is to apply a filter 0.01 to 0.1 Hz and see if there is a clear signal. A more quantitative test is to make spectral analysis of the S and/or surface waves to observe where the signal to noise ratio approaches 1. There are examples of inversions of small near events (m=2−3) with frequencies as high as 3 Hz. However, this is not what generally can be expected to work. It is possible to use any or all of the components Z, R and T, however the T-component is often the most important. It is recommended to use at least 4 stations with at least 6 seismograms, however that is rarely enough to get a reliable solution. Real signals are often more complicated and of longer duration than the synthetic signals, so it is easier to fit signals with a few simple pulses.

Use velocity or displacement
Our tests show that it does not make much difference for good data. In some cases it might be easier to get stable velocity traces than displacement traces since the effect of the enhancement of low frequency noise from the conversion to displacement is avoided. Whatever is selected in MULPLT is what is used by all programs. There is a check that there is no discrepancy between the data and the Green’s function in terms of displacement and velocity.

Technical steps to do MT
**Step 1** Select channels to analyze and write out a rotated instrument corrected data file in Helmberger format which is used to make input files to the Dreger program. Note, that all filters used must be 4 pole bandpass Butterworth filters, either one way or two ways. Although MULPLT can generate other filters, they can currently not be used for MT inversion.

- Make sure the event is well located and all response files are available. If doubt about depth, fix it in S-filer header line.
- Start EEV.
- Plot data filtered, it is easier if filter is fixed.
- Select channels desired for analysis. Select as many channels as possibly can be used. It is possible later to deselect without going back to MULPLT. NOTE: You can only make a data set of the channels seen on the screen! See below how to deal with many channels.
- Select a time windo starting about 30 s before the first P and lasting for about 10 min, but at least long enough to get all waves to use for analysis.
- Rotate channels (if 2 horizontals), check that all channels are rotated (no back azimuths of 999).
- Select filter band and plot signals instrument corrected and filtered. Start with just filtering the signals e.g. in band 0.01 to 0.1, this gives an idea of which channels have good low frequency signals. A good filter to use is often 0.03 to 0.1 Hz. For small events, a filter of 0.5 to 1.0 might have to be used. Inspect the signals to see if they look reasonable. E.g., the instrument corrected amplitudes should not be very different. At this stage errors in the response files or bad data might be detected. If so correct data and select again. A wrong response can seriously affect the solution.
- When signals look ok and are displayed instrument corrected, filtered and rotated, press button OutW, and wait for message in to right hand corner 'File mulplt.wav finished'. The Ascii file mulplt.wav has now been written out. It can take some time since it is an Ascii format of real numbers (see below). In a multi-trace window there can be missing data in front of the signals and at the end of the signals for some channels. These gaps are filled out with the DC level.
- Quit MULPLT.

**How to deal with many channels**: By default, MULPLT shows 99 channels per screen, but this is often reduced to a smaller number by setting parameter NCHAN PER SCREEN in MULPLT.DEF to a number like 24. So if e.g. 100 channels are available, the data will be shown on several screens. The procedure is then:

- Select channels to use on each screen. If no channels are selected on a screen, all will be used.
- When the selection is finished and if there are more channels than fitting on one screen, press N. This will plot all channels on one screen and a file with all channels desired can be made. Pressing N again brings back the original number of channels per screen.

At this stage, a file with possible data for analysis has been written out with the original sample rate, instrument corrected (units nm or nm/s) and filtered. The two first letters in the component names has been replaced by RR to indicate reprocessed data. The file can be plotted with MULPLT or from EEV with command pd.

**Step 2** Create parameters needed for MT
The generation of the Greens function needs a series of parameters including the crustal model, which might be the most critical input. The model used will be taken from the \texttt{STATIONx.HYP} file so it will be possible to use a station file different from the default by either working in a local directory with a local \texttt{STATION0.HYP} or having a \texttt{STATIONx.HYP} in DAT, where \textit{x} corresponds to the model ID given in the S-file. Note that a \texttt{STATIONx.HYP} file can contain Q and density, but often does not and then default values are used. All parameters will be written in the S-file in the SYNT format (including extra mt-variables), see section on synthetic seismograms and the example below. The stations selected for analysis will be the stations given in the \texttt{mulplt.wav} file as made under step 1. Default values are given for many parameters. In order to generate parameters, in EEV:

- Give command \texttt{mtp} (p for parameter). All needed parameters are now stored in S-file as well as parameters for the synthetic seismogram programs, since many of these parameters are the same. An example of the parameters is given below. At this time a backup file of \texttt{mulplt.wav} is made for future reference. The name is \texttt{yyyy-mmdd-hrmm-ss.mulplt.wav}.
- Edit the S-file to change default parameters to desired values, see example below. Often a first test can be made with the default values. By default, modeling is only done for one depth, but it is possible to test a range of depths by editing the parameter file. It is however recommended to use the default depth from the S-file for the initial test.

\textbf{NOTE:} Command \texttt{mtp} does not overwrite any parameters already in S-file. If a completely new set of parameters is needed, all the old ones can be deleted in the S-file or by using command \texttt{mtd}.

\textbf{Step 3} Generate and inspect unfiltered Greens functions.

The Greens functions should now be generated with the parameters given in S-file. In EEV:

- Give command \texttt{mtg} (g for Green’s functions). This makes Helmerger format files \texttt{all.green} with all the time series Greens functions needed for the given data set and the requested depths \texttt{green}. By default only one depth is used. This might take some time (minutes) depending on number of points used, however the time is independent of the number of stations used. The default is to generate a 512 point time series which by default is 512 secs long. \textbf{NOTE:} All previous Greens function files are deleted before the new ones are made.
- Plot the Greens functions with command \texttt{pg} (g for Green). It is useful to check if the Greens functions look "reasonable". \textbf{NOTE:} If Greens functions for several depths have been made, only the Greens function with the largest depth can be plotted in this way (file \texttt{all.green}). The other ones must be plotted directly with MULPLT outside EEV. For component codes, see Dreger documentation in SEISAN. Note that the transverse components TSS and TDS have no P-waves so they appear to start later. Not all models and distances might produce reasonable signals, there should be some resemblance with the data signals. Note that the signals, starting at the origin time (read from S-file) have been time shifted with the reduction velocity to appear to arrive at similar times. At this stage it might be decided that a different time window or sample rate is needed. Then edit S-file and redo step 3.

\textbf{Step 4} Decide on time window length for analysis.

The Green’s functions are typically generated for 512 secs (sample rate 1.0) and typically a smaller window around the signals is used like 200-400 s (default 257 s). From the Greens functions signal, it can be seen how long a minimum window is needed. The window should be longer than the signals. Before inversion, the signals are also time shifted (like the Greens functions are already) with the reduction velocity in order to make the data file smaller.
• Edit S-file and adjust parameter MT-NP-USE to desired length.

Note that all the Green’s function files and the time shifted data files do not have accurate absolute time due to time shifting.

Step 5 Make the inversion

The desired time windows from all.green are filtered by the selected filter and written out. The filtered time shifted window from mulplt.wav are now selected, filtered by a (desired sample rate)/5 Hz antialias filter and resampled to the desired sample rate and written out. The inversion is now performed. All this is done in EEV. NOTE: All data files from previous inversion are deleted before the data selection.

• In EEV, give command mti (i for inversion).

The inversion is now done and the results given on the screen. If a range of depths have been selected, inversion will be done for each depth and the inversion will be repeated for the depth with the best fit so this becomes the last inversion, which can be save in S-file. A table of fit parameter (VR) and depth is displayed together with corresponding fault plane solution. If results are good (see later), they can optionally be saved in the S-file. Note particularly the value of Zcor which is how many samples the data has to be shifted to fit with the Greens function.

Step 6 Check results

See Dreger documentation in INF for the explanation of the output. The variance reduction (VR) is shown for each station as well as for all stations. A low value indicates a bad fit and a negative VR might indicate inverted polarity. The most important check is to see how the synthetic seismograms fit the observed seismograms.

• In EEV, give command pm. The plot of overlaid seismograms is now shown. They should be similar. Plot also with fixed scale to see the absolute difference between the traces.

The fault plane solution can be plotted (if saved in step 5) with command fo. Compare to any solution from other sources (if given in S-file, see section on fps in SEISAN).

In the above plots, the original data has been shifted corresponding to the value of Zcor. This means that, for a positive Zcor, the last Zcor data sample on the trace has no data and is replaced by zeros. Similarly if Zcor is negative the first Zcor samples are zero.

Judging the results

See the Dreger documentation for a discussion. Generally the variance reduction VR should be as high as possible. A bad fit could give an unrealistic moment (and Mw) so that is also an indicator of the quality. A good fit is not a guarantee for correct results. If e.g. the gap is large, there might not be sufficiently different data to give a reliable solution, even if the fit is good. The quality given has been assigned by Dreger as follows

\[
\begin{array}{c|c}
0 < VR < 20 & \text{Quality} = 0 \\
20 < VR < 40 & \text{Quality} = 1 \\
40 < VR < 60 & \text{Quality} = 2 \\
60 < VR < 80 & \text{Quality} = 3 \\
80 < VR < 100 & \text{Quality} = 4 \\
\end{array}
\]
Figure 26.1: Plot of original filtered instrument corrected data (blue) compared to the synthetic seismograms (red). The filter used is 0.02 to 0.05 Hz. The plot is made with a fixed scale of 30 000. Note how the T-components dominate the solution. The data is the Dreger test data included in the SEISAN training data. Note that when plotting a file in Helmberger format, the overlay function (see MULPLT section) is turned on automatically for channels starting with SY.
CHAPTER 26. WAVEFORM INVERSION

Figure 26.2: Same data as shown in previous figure but only for station BKS. The data is now auto scaled. The fit on all channels is quite good. Notice the small amplitude of the radial component.

It is necessary to check the solution against the P polarities to confirm that they generally match the solution as wrong alignment can result in inverted solutions. However, it is not to be expected that all the synthetic polarities fit the observed polarities since the fault plane solution from MT (measures overall slip) might be different from fault plane solution with polarities (measures the slip of the initial rupture). Use command fo to plot the solution together with observed polarities recorded in the S-file.

Example of a run of the inversion

```
# 2 12 Aug 1998 14:10 25 L 36.755-121.462 8.00FF .50 BGS 4 ? mti

***Event to invert***: C:\Seismo\REA\TEST_\1998\08\12-1410-00L.S199808

Inversion in: displacement
Number of stations: 4
Number of points to use: 250
Depth: 8.0
Sample rate: 1.000
Reduction velocity: 8.0
Filter: 0.020 0.050

Skip for down sampling for PKD 20
Skip for down sampling for BKS 20
Skip for down sampling for CMB 20
Skip for down sampling for KCC 20
```
26.1. MOMENT TENSOR INVERSION IN SEISAN

PKD  Distance(km)=  122.0  ShiftVel(s)=  0.1  Offset(sample)=  0
BKS  Distance(km)=  142.0  ShiftVel(s)=  2.5  Offset(sample)=  0
CMB  Distance(km)=  171.0  ShiftVel(s)=  6.2  Offset(sample)=  0
KCC  Distance(km)=  201.0  ShiftVel(s)=  9.9  Offset(sample)=  0

Writing PKD__.green  ShiftVel: Shift in s due to reduction velocity
Writing BKS__.green  Offset: Zcor parameter in S-file
Writing CMB__.green
Writing KCC__.green

Output from tdmt_invc:  See Dreger manual for following output

Depth=8
Station Information
Station(0):  PKD__.data  R=122.0km  AZI=137.0  W=1.000  Zcor=14
Station(1):  BKS__.data  R=142.0km  AZI=331.0  W=1.164  Zcor=13
Station(2):  CMB__.data  R=171.0km  AZI=34.0  W=1.402  Zcor=13
Station(3):  KCC__.data  R=201.0km  AZI=71.0  W=1.648  Zcor=12
M0=4.12984e+023
Mw=5.0
Strike=223 ; 130
Rake=18 ; 172
Dip=83; 72
Pdc=98
Pclvd=2
Piso=0
Station(0)=78.607796  4.95437e+010
Station(1)=82.944862  3.37622e+010
Station(2)=82.605667  1.91152e+010
Station(3)=56.903259  6.36377e+009
VAR=7.48972e+006
VR=79.39 (UNWEIGHTED)
VR=79.00 (WEIGHTED)
Var/Pdc=7.668e+004
Quality=3

Update event with new mt solution(n=enter/y) ?

How and where parameters are changed for making tests

After the first parameter file in the S-file has been made, these parameters can only be changed by manually editing the S-file or deleting part or all of the parameters in the S-file and running command mtp again. Possible changes:

- New model: Delete in S-file, edit station file and run mtp. Or correct directly in S-file. However if a new model results in new distances and azimuths, then the station lines should also be deleted before using mtp. Similarly with the depth line if depth changes.
- New depth: If location is unchanged, only change depth in parameter file and start with step 3.
- Range of depths: Edit S-file depth line.
- New event location: Delete station lines in sfile and start with step 2.
- Take out some stations: Change the ID in the first station line (the one with mt specific parameters), e.g. write xSTATION instead of STATION. Then run inversion again, step 5.
- Take out components for a station: Edit the field MT-COMP: TRZ, see below. Then run inversion again, step 5.
- Change filter: A new data selection in MULPLT must be made with the new filter and the new filter must be written into the S-file. There is a check if filter in S-file corresponds to filter in data file. Then continue with inversion, step 5.
- Add a completely new station: Start from the beginning, this requires complete data selection and computation of Green’s functions. The simplest is to delete all parameters with mtd.
- Change from displacement to velocity: Start from the beginning.
- Change sample rate: Change in S-file and start from step 3 making Greens functions. Remember to change the number of points for inversion correspondingly, that is e.g., if sample rate is doubled, number of points must also be doubled to analyze the same length time window.
- Change reduction velocity. Change in S-file and ideally start from step 3. However, starting from step 4 gives the same result provided the time window is long enough, see also below.
- Adjust time shifts: Zcor gives the number of samples the data automatically has been shifted to correlate with the Greens function. Check how the synthetic seismograms fit the observed seismograms. This number can be adjusted in the s-file. The number should be Zcor plus or minus a few data points. If zero, the automatic correlation is used. If real data is seen to the left of the synthetics, reduce Zcor and increase if the data is to the right of the synthetics (indicated on top of plot as Data left. -Zxor). Zcor can be positive or negative.
- In general, if deleting a line with parameters, they can be regenerated by command mtp.

The most common editing in the S-file can be done with command mte in EEV. The command shows:

```
Edit MT station lines
NUM STAT ACTIVE COMP ZCOR-SFIL ZCOR-AUTO DIST AZ
 1 PKD 1 TRZ 0 40 122 137
 2 BKS 0 TRZ 0 0 142 331
 3 CMB 1 TRZ 0 39 171 34
 4 KCC 1 TRZ 0 38 201 71
Enter station number to flip if station is used or not
Enter station number and components to use, e.g. 4 TZ
Enter station number and new zcor, e.g. 3 33
Enter to terminate
```

With this command it is possible to very quickly weight out or in a station, select components and change zcor.
If a range of depths is used or the new mt solution is updated, the mt is written to a file named `psmeca.in` that can be used to plot the mt solution with the GMT program `psmeca`. With the command

```
psmeca psmeca.in -R10/80/0/42 -JX16/-20 -Sd0.3 -Gred -P -B10f5:"Variance reduction":/1:Depth:
> mt.ps
```

the double couple part was plotted in figure 26.3. Using `-Sm0.3` will show the mt.

**Summary of mt related commands in EEV**

- `mtp` Make parameters
- `mtd` Delete all parameters
- `mte` Edit parameters in S-file
- `mtg` Make Greens functions
- `mti` Make inversion
- `pm` Plot observed and synthetics
- `pd` Plot mulplt.wav
- `pg` Plot greens functions
- `fo` Plot fault plane solutions

**Influence of parameters**

*Filters:* Try to find a filter giving a good signal to noise ratio. There can be substantially difference using different filters.

*Reduction velocity:* It has little influence on the results except that Zcor changes due to relative change in arrival times. It is normally selected to include signals before the P arrival and to include all surface waves. Even different reduction velocity for real and Greens function data does not matter. The use of reduction velocity only has the purpose of reducing the length of the traces by putting events close in time. This is particularly important when using events in with a wide distance range.

*Time window:* The most important is that the time window includes the whole time signal to be inverted. The time window actually used by the program will depend on how close it is to a $2^n$ number. The correlation is done only with $2^n$ samples. The number of samples selected for correlation can be both larger and smaller than the number of samples in the data file. E.g. if number of samples is between 90 and 181, 128 samples will be used and if between 182 and 362, 256 samples is used etc. The inversion also seems to use a $2^n$ number and there can, in a few cases, be a radical difference between using e.g. 256 and 257 points in calculating the fit VR, but apparently not the solution itself. The default number of samples to use is therefore 257. If the number of samples is near a $2^n$ number, use a number a bit larger than $2^n$. This change in fit is not generally observed, in most cases VR is not affected. Inspect the Greens function file all.green (command `pd`) or use MULPLT with all.green$$ to see how long the Green function signal is and similarly look at the data files. The time shifted data files are STAT.data and can be plotted with MULPLT.

*Sample rate:* Seems to have little influence if well above the frequencies of analyzed data. One or two Hz seems ok for data where the inversion is made for frequencies below 0.1 Hz. For 1 Hz data 4−8 Hz or similar (depending on sample rate of original data).

*Number of stations and components:* It is often difficult to get good results with all stations and components. Start with a few stations (even one god one) and gradually add data. Note that changing station
configuration also can change the time shift between synthetic seismograms and the data. It is important
to have as small a location gap as possible.

Zcor: Zcor is calculated by correlating the Greens functions with the data and the synthetic seismograms
might need a correction as observed from the overlay seismograms. The Zcor time shift will change
with different solutions so there is no final Zcor that will work in all cases. Small changes in Zcor can
significantly improve the results. In the worst case, Zcor might have to be large enough to reverse the
polarity of a signal or even larger if e.g. P has been correlated with S (small events at higher frequencies).
The automatic correlation is what creates most problems.

What can go wrong

- Very bad fit. This can be caused by the correlation not working well, Signals might be shifted
several cycles. Try using a different time window, particularly a longer one. This can also be cause
by a bug which results in a factor 2 wrong sample rate. it is clearly seen when comparing the
synthetics and real data. Just run again usually fixes the problem:

- Crash of inversion program. This might be caused by correlation not working. Zcor for a particular
station might have a value of millions. Do not use the particular station.

- Deselecting components does not always seem to work, then use all 3 component or deselect station.

- The invasion does not start. Use Ctrl+c and start again.

Re-running a previously analyzed event

The S-filer contains all a parameters used including the start time and duration of the data file so it
is possible to extract the same data file for the same channels again. The Green’s functions must be
regenerated by command mtg. If the user does not delete the backup files (see above) the data file is
available as backup file and a question will be given if the previous data file should be used.

Running the programs independently of EEV

Once a first run has been made, the programs can be run independently of EEV using the original
parameter files. This can be an advantage if the user wants to change some of the hardwired parameters.
However, this can only be done for one depth.

FKRPROG:SEISAN: The only parameters which might be tested are the group velocities. Dreger is vague
about what they should be but their values seem to have some influence on the Greens function. It is
also possible to edit other parameters independently like the model.

- Edit the green.par$$$. file ($$$ is depth).
- Run program fkrprog_seisan: fkrprog_seisan < green.par$$$
- Convert to time domain by running vwint_seisan

TDMT_INV:SEISAN: The only parameter that cannot be tested through EEV is to make the analysis
window smaller than the data window. This sometime improves the results.

- Edit parameter file mt_inv.in
- Run TDMT_INV: tdm_inv_seisan
Running in this way, the synthetic file readable by SEISAN is not generated and cannot be updated with the results.

**Example of parameters in S-file**

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNT:</td>
<td>MODEL--:</td>
<td>THICK VP VS DENS QP QS 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MODEL--:</td>
<td>1.000 3.200 1.500 2.280 600.000 300.000 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MODEL--:</td>
<td>2.000 4.500 2.400 2.280 600.000 300.000 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MODEL--:</td>
<td>1.000 4.800 2.780 2.580 600.000 300.000 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MODEL--:</td>
<td>12.000 6.210 3.400 2.680 600.000 300.000 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MODEL--:</td>
<td>8.000 6.890 3.980 3.000 600.000 300.000 N 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MODEL--:</td>
<td>60.000 7.830 4.520 3.260 600.000 300.000 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MODEL--:</td>
<td>50.000 8.000 4.600 3.260 600.000 300.000 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MODEL--:</td>
<td>12.000 6.210 3.400 2.680 600.000 300.000 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MODEL--:</td>
<td>8.000 6.890 3.980 3.000 600.000 300.000 N 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MODEL--:</td>
<td>60.000 7.830 4.520 3.260 600.000 300.000 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MODEL--:</td>
<td>50.000 8.000 4.600 3.260 600.000 300.000 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>ST-D-RK:</td>
<td>48.0 85.0 -1.0 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>DEPTH--:</td>
<td>10.0 5 5.0 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>NPPOINTS:</td>
<td>512 MT-NP-USE 280 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>TIMES--:</td>
<td>60.000 INITIAL 0.000 SY-TRACE 60.000 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>BOUPAR--:</td>
<td>800.0 2000 0.010 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>PHASES--:</td>
<td>Pg Sg PmP SmS SmP 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>DT-Tsou:</td>
<td>0.050 .100 MT_RATE 1.0 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MTSTART:</td>
<td>1998-0812-0410-00.0 MT-WINDOW 305.0 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MT-D-V--: displacement 3</td>
<td></td>
</tr>
<tr>
<td>SYNT:</td>
<td>REDVELO:</td>
<td>0.0000 MT-REDVL: 8.0 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>COMPON--:</td>
<td>RADIAL 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>STAT-AT:</td>
<td>no 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>MT-FILT:</td>
<td>0.015 0.030 4 1 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>NSTAT--:</td>
<td>4------------------------------------------------------3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>STATION: PKD S Z DISTANC:</td>
<td>122.0 MTOFFSET: 22 MT-COMP: TRZ 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>STATION: PKD AZIMUTH:</td>
<td>137.0 317.8 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>STATION: BKS S Z DISTANC:</td>
<td>142.0 MTOFFSET: 0 MT-COMP: TRZ 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>STATION: BKS AZIMUTH:</td>
<td>331.0 151.0 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>STATION: CMB S Z DISTANC:</td>
<td>171.0 MTOFFSET: 0 MT-COMP: TRZ 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>STATION: CMB AZIMUTH:</td>
<td>34.0 214.1 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>STATION: KCC S Z DISTANC:</td>
<td>201.0 MTOFFSET: 0 MT-COMP: TRZ 3</td>
</tr>
<tr>
<td>SYNT:</td>
<td>STATION: KCC AZIMUTH:</td>
<td>71.0 252.3 3</td>
</tr>
</tbody>
</table>

Most of the parameters are explained under synthetic seismograms. The new ones used only for mt and other important ones are:

**DEPTH--**: The first number is start depth, the following number of depth to test and the last number is the increment in depth. The default is one depth only. **MT-NP-USE**: Number of points for the inversion, default 257. The number does not have to be $2^n$ but it seems that, in some cases, a number a bit larger than $2^n$ is better than $2^n$ or a bit smaller.

**NPPOINTS**: Number of points in time domain used to make Greens function, default 512. This number must be $2^n$.

**MTSTART**: Start time of data window used. **MT-WINDOW**: Length (s) of data window used. **MT-RATE**: Sample
rate to use. This rate will be used for Greens function generation and the observed data will be down-
sampled to this rate. *NOTE: The rate must have value so only skipping samples in the data can be done.*
So the rate 1 can nearly always be used while rate 3 rarely can be used. The time window for the Greens
function will then be NPOINTS/MT-RATE. Default 1.0 samples/s.
**MT-REDVL**: Reduction velocity, default 8 km/s.
**MT-FILT**: Filters to use for both data and Greens functions.
**MTOFFSET**: Offset in samples for the data relative to the Greens function. Default 0.
**MT-COMP**: Indicate which component to be used. By default all 3 are used, but any combination can be
selected. T, R and Z can come in any order but must be within column 76:78.

**Technical notes**

The Dreger MT inversion essentially consists of a Green's function generation program, *fkrprog* (in
SEISAN called *fkrprog.seisan*), several data manipulation programs and scripts using SAC to prepare
data for the inversion program *tdmt_invC* (in SEISAN called *tdmt_invC.seisan*). The two key programs
have been left nearly unchanged (all changes are clearly marked in programs) while the data manipulation
programs mostly have been replaced by standard SEISAN functions (mostly within EEV) and one
modified Dreger program *wvint9* (now called *wvint.seisan*) so only 3 programs have been added to
SEISAN. A significant change is that Dreger uses cm as a unit while SEISAN uses nm, so software has
been changed to nm like elsewhere in SEISAN. The Dreger code has no provision for using less than 3
channels. However, undocumented information indicates that if a data channel has zeros, it is not used
and this is how it is implemented in SEISAN. In some cases it does not seem to work well, should be
investigated more. The data and Greens functions in Dregers software are using the Helmberger format,
a simple Ascii format without reference to time and channel name. SEISAN will, for simplicity use the
same format, but it has been extended to also include channel information, absolute time and an ID of
the event being processed (the S-file name), see example below.

```
3 0.020 0.050 4 1 C:\Seismo\REA\TEST\_1998\08\12-1410-00L.S199808
(7e14.5) displacement
  0.0000e+00  0.0000e+00  0  0  0.00
120  0.100  0.0000e+00 PKD RR T 1998 812 1410 5.983
0.48303E+01 0.48328E+01 0.48315E+01 0.48305E+01 0.48293E+01
0.48275E+01 0.48264E+01 0.48257E+01 0.48247E+01 0.48237E+01
0.48236E+01 0.48208E+01 0.48204E+01 0.48195E+01 0.48179E+01
0.48153E+01 0.48138E+01 0.48090E+01 0.48077E+01 0.48043E+01
```

The two first lines are main headers. The first line has number of channels in file (format i8) and has been
extended with, filter band, number of poles and passes and S-file name. Only the number of channels is
required information for plotting. The second line gives format of data, extended with help text which in
this case is information that file is in displacement (nm). The following two lines are channel headers for
each channel. The first channel header has undocumented information. The second channel header has
number of samples in channel (i8), sample interval(s)(f10.3), undocumented and starting from column
32, the station code, channel code and start time.

The program flow in SEISAN is

- Plot, rotate, filter and instrument correct signals. This generates an output file *mulplt.wav*. A
  parameter is set in MULPLT to enables the output. This file has not been re-sampled.
- Make parameters in S-file (command *mtp*): Parameters are selected from *mulplt.wav* (stations to
  use, filter and whether displacement or velocity), from s-file (depth, distances and azimuths), from
station file (model) and some are hardwired. The operation takes place in EEV. At this point the backup data file is made.

- Generate the 10 Greens functions (command mtg): All old Green's functions are deleted and parameter files to \textit{fkrprog.seisan} (\textit{green.par}$$\$, $$\$\$ is depth) are made from the S-file (inside EEV), \textit{fkrprog.seisan} is run giving an output file with frequency domain Greens functions \textit{green.out}. This file is converted to a Greens function time domain files, \textit{all.green}, with \textit{wvint.seisan} (modified version of original program \textit{wvint9}). Since it is assumed there is no explosive component, only 8 of the 10 greens function are written out in time domain (as the original program). \textit{all.green} is in Helmberger format. Information of using displacement or velocity is read from S-file.

- Make inversion (command mti). In EEV, same length time windows, one for each station, are extracted from \textit{all.green} and \textit{seisan.mt.out}. Names are \textit{STA.green} and \textit{STA.data}. The old \textit{STA.green} and \textit{sta.data} are deleted first. The Greens function files are antialias filtered with a LP filter of 4 poles at (desired sample rate)/5 Hz (not sure it is needed), re-sampled and time shifted, all in EEV. If a channel has been deselected or not available, zeros are written out. The Greens functions are band-pass filtered and anti alias filtered with same filters as used for the data. The parameter file (\textit{mt_inv.in}) for the inversion program \textit{tdmt.invc.seisan} is made (in EEV) and the inversion program started from EEV. The inversion program makes an output file with signals and synthetics (\textit{synt.out}) which is converted to Helmberger format file \textit{synt.seis.out} in EEV. Command \texttt{pm} will plot this directly in EEV. File \textit{mt_redi.out} (as the original) has details of the results and is read by EEV to optionally get results back into the S-file.

\textit{Fkrprog.seisan}: The parameter file format has been changed to include station names and the s-file name.

\textit{Tdmt.invc.seisan}: The program has minor changes to accept new Helmberger format (it also reads the original format). It did not work properly under Windows with more then 450 points, so memory allocation was doubled to fix this. The plotting routine has been simplified to only write original and synthetic seismograms.

To get correct overlay, the channels must be sorted alphabetically. MULPLT turns on sorting (normally set in MULPLT.DEF) if the filename is \texttt{synt.seis.out}.

The SEISAN implementation is dimensioned to max 99 stations.
Figure 26.3: The double couple part of the moment tensor solution shown with respect to the variance reduction at different depths. The GCMT solution is also shown with arbitrary variance reduction. This figure was made with the GMT program psmeca, see text.
Chapter 27

Calculation of coda q, CODAQ, and related programs

This section contains the main program CODAQ to calculate coda Q, program CODAQ_AREA to sort output from CODAQ into areal bins, CODAQ_POLYGON to average in a polygon and a help program AVQ to average Q-relations.

Codaq

The program will calculate coda Q (hereafter called Q) for a series of events and stations at given frequencies. On completion, the average values are calculated and a Q vs f curve is fitted to the calculated values. The program will also plot the individual events and filtered coda windows.

The principle for calculation is the standard coda Q method, whereby a coda window is bandpass filtered, an envelope fitted and the coda Q at the corresponding frequency calculated. The envelope is calculated RMS value of the filtered signal using a 5 cycle window. The program used here is the one described in Havskov et al. [1989] and in more details in Havskov et al. [2016]. The program can only operate in connection with the SEISAN format S-files. The program can use all waveform file types accepted by SEISAN and there can be more than one waveform file in the S-file. The program will also take advantage of the SEISAN database structure.

NOTE From SEISAN version 10.6, the error calculations in CODAQ has changed. Luis Matias pointed out that they seemed too small and indicated how to do it correctly (see https://en.wikipedia.org/wiki/Propagation_of_uncertainty). Consequently, most errors have increased and this also affected the calculated of Qzero and Qalpha (see later for definition). So rerunning CODAQ with identical data might give different results from version 10.6 as compared to earlier versions.

27.0.1 Input

The calculations are controlled by a parameter file called codaq.par and the actual event-station combinations to use are given in codaq.inp. Example files are in DAT, and with the test data set and the example files in DAT, a test run can be done. An example of a parameter file is shown below:

-------------------------------------------------------------------
CHAPTER 27. CALCULATION OF CODA Q, CODAQ, AND RELATED PROGRAMS

start in s-times and Vp/Vs ratio (optionally)
2.0
absolute start time (sec)
0
window length (sec)
20
spreading parameter
1.0
constant v in q = q0*f**v
1.0
minimum signal to noise ratio
2
Noise window in front of signal and length of RMS noise window
15,5
minimum correlation coefficient
0.50
maximum counts to use
64000
number of frequencies and number of octaves (optional)
3
frequencies and bands (optional if octave given)
4,2
8,4
16,8
default stations(1. line) and components (2. line), 30a5
HYA ASK SUE
S Z S Z S N
-------------------------------------------------------------------

Start in s-times and Vp/Vs ratio (optionally): Normally the coda window starts at twice the S-travel time from the origin, this factor can be varied and might be chosen differently in special cases. Note that the S-time is calculated from the P-time so a P-time must be present. This also means that if a Pn is used, the coda window will start at 2 times the Sn travel time, which might be substantially different from 2 times the Sg travel time. e S-time is calculated from the P-time using Vp/Vs = 1.78. Optionally, the user can select an Vp/Vs ratio to be used. This parameter is optional so parameter files prior to version 8.3 can be used.

Absolute start time: If 0.0, above parameter is used. However if different from zero, an absolute start time relative to the origin time is used for the start of the coda window. This might be useful since different start times (meaning different lapse times) might produce different q-values. To use this parameter, one must be certain to choose it long enough which can be checked with the plots. If the absolute start time is smaller than (Start in s-times) multiplied by the s travel time, the station will be skipped and a message given.

Window length: This is the coda window length in secs. Use at least 20 secs to get stable results.

Spreading parameter: The geometrical spreading parameter used in q-fit, normally 1.0 is used.

Constant v in q = q0 * f**v: For all q(f) values, q0 is calculated using a fixed v, use e.g. 1.0. This parameter has no influence on the individual q calculations.
Minimum signal to noise ratio: In order to accept a \( q \) value for the average, the signal to noise ratio must be above this value. The signal to noise ratio is calculated using the last tRMS (see next parameters) secs of the filtered coda window and the first tRMS secs of the data file window. If the data file starts with noise or in the P signal, the s/n ratio will be in error. A reasonable value is 5.0.

Maximum counts to use: If the count value in a coda window is above this value, the window is not used. The intention is to avoid using clipped values. From SEISAN version 7.2, there is also an automatic checking for clipped values in addition to ‘maximum counts’.

Noise window in front of signal and length of noise window, tnoise and tRMS: The first number is the number of seconds of noise to plot in front of the signal. In previous versions, 15 secs was hardwired, but sometimes there was not 15 secs of noise before the P. The second number is the length of the noise window used for calculation of the signal to noise ratio. This was earlier hardwired to 5 secs.

Minimum correlation coefficient: In order to use the \( q \) value in the average, the correlation coefficient of the coda \( q \) fit must be larger than or equal to this value. NOTE. Correlation values are in reality negative, but are always referred to as positive in the following. An acceptable value depends on the data, try to use a value higher than 0.5 (in reality -0.5)

Number of frequencies and number of octaves: Number of frequencies to use, maximum 10, 5 is a good number. The number of octaves for the filter can also be given, then all filters have the same bandwidth.

Frequencies and bands: The corresponding center frequencies and frequency bands. The frequency band should increase with increasing frequency to avoid ringing. E.g. 8,4 means that the signal is filtered between 6 and 10 Hz. It is advisable to use constant relative bandwidth filtering, to get an equal amount of energy into each band. The relative bandwidth is defined as \( RBW = (f_u - f_l)/f_o \) where \( f_u \) and \( f_l \) are upper and lower frequency limit respectively. Such a filter would be e.g. 4 ± 1, 8 ± 2. 16 ± 4. The frequency representing the energy in a particular filter band, is the geometric center frequency calculated as \( f_c = sqrt{f_u f_l} \). Since the user probably wants to calculate coda \( Q \) at the given frequency, the normal option (new in SEISAN7.2) is that \( f_u \) and \( f_l \) are calculated such that the given bandwidth (e.g. 4 Hz) is used, but the actual \( f_u \) and \( f_l \) will give the specified central frequency. It is still possible to calculate as before, where \( f_u \) and \( f_l \) will be exactly as specified (but the geometrical center frequency will not correspond to specified center frequency) by giving the bandwidth as a negative number. The filter width should be at least one octave like 2-4 and 8-16. The number of octaves can be given with number of frequencies so then there is no need to calculate the correct bands. It is recommended to use a 1-2 octave filter and when using the option of specifying octaves, the geometric center frequency is always used. At ther end of the calculations, the octaves and filter bands is written out. Default stations: The stations and components that will be used if not specified in the \texttt{codaq.inp} file. The stations are on the first line and the components on the second line. A maximum of 300 channels can be used. THE 2 LINES MUST BE THERE CONTAIN AT LEAST SOME BLANK CHARACTERS, if not, stations will not be read from \texttt{codaq.inp} file and the program will crash.

Use of components; Component blank: Note that the program will use the first available component for the given station in waveform file. The component actually selected will be shown in output and on plot. Component with orientation e.g. Z: First Z-channel for station is selected.

After reading the parameter file, the program will by default use the \texttt{codaq.inp} file to get the event station information. However, any other name can be used if specified interactively, see below. The station codes can have up to 5 characters. The \texttt{codaq.inp} file will consist of a series of lines each giving an event identifier (an INDEX file). An easy way to generate the file is using the SELECT program. The file can also be generated with EEV using the (C)opy option making a file called indexeev.out. An example, where default stations are used, is shown below:
Since the above example only uses the default stations given in \textit{codaq.par} there are no stations in the input file. Below is an example where particular stations and components have been selected with particular events, for this to work the station line and component line in \textit{codaq.par} MUST be blank. Component lines can be blank or only with orientation.

\begin{verbatim}
1 /top/seismo/seismo/REA/BER__/1992/06/16-0343-38L.S199206
  HYA KMY BER ASK TRO
  S Z S E B E S Z S Z
3 /top/seismo/seismo/REA/BER__/1992/06/16-1311-58L.S199206
  HYA
  Z
7 /top/seismo/seismo/REA/BER__/1992/06/30-1504-30L.S199206
  HYA EGD
  S E Z
\end{verbatim}

Note that the numbers to the left originate from the index file and do not have any importance. The long name with the directory structure, is the name of the pick file (S-file) in the database, if the S-file is in the local directory, it can have just the event id, in this example starting with 30-....The waveform file name is in the S-file. Following the S-file name is, (like in the parameter file), first a line with station codes followed by a line of component codes. Like in the parameter file, if a component is not given, the first component with given station name will be chosen. THE COMPONENT LINE MUST BE THERE, EVEN IF BLANK. Since it can be quite a lot of work to generate this file manually with both stations and components, SELECT has an option to generate it, see SELECT. SELECT is able to make an output file with all event-station combination within a given range. However, SELECT will give incomplete component names since not complete in S-file. It is possible to use one or three components. The file from SELECT is called index.codaq.

Below is an example of a \textit{codaq.inp} file, where it is assumed that the S-files are the current directory. This file can also be generated with DIRF.

\begin{verbatim}
16-0343-38L.S199206
  HYA KMY BER ASK TRO
16-1311-58L.S199206
  HYA S E
30-1504-30L.S199206
  HYA EGD S N S E
\end{verbatim}

In the above examples, all results from one run are averaged. However it can sometimes be desirable to run several datasets and get individual results from each. There is therefore the option of running one set of events with many different sets of stations and for each set, the results are written out separately. Typically in a large area, one would want to get \textit{codaq} for each station or small groups of stations using a large earthquake set. This can be done using several sets of default stations, see example below.
In this example, the first dataset has two stations, the others only one. The summary output for each dataset is given in file codaq.summary. An example is seen below, for abbreviations, see later:

SUE S Z KMY S Z
ntotal=37 q0=71 sd=25 v=1.19 sd=0.14 cor=0.99
ASK S Z
ntotal=10 q0=54 sd=22 v=1.19 sd=0.19 cor=0.98
ODD1 S Z
ntotal=11 q0=61 sd=19 v=1.25 sd=0.13 cor=0.99
SUE S Z
ntotal=20 q0=54 sd=26 v=1.27 sd=0.19 cor=0.98

Average coda Q is also calculated for each station or channel. The default is to calculate average for all components for each station. The output is given in file codaq.channel and also with more details in codaq1.out. Optionally the averages can be made for each channel if codaq is called with the argument -c. An example of an output file with channel averages is seen below.

PIL BH Z n=4 q10=1066 q0=147 sd=5 v=0.86 sd=0.01 cor=1.00
PIL BH N n=5 q10=1012 q0=87 sd=25 v=1.06 sd=0.12 cor=0.99
PIL BH E n=5 q10=995 q0=66 sd=9 v=1.18 sd=0.08 cor=1.00
YJI BH Z n=11 q10=1033 q0=113 sd=5 v=0.96 sd=0.03 cor=1.00
YJI BH N n=3 q10=965 q0=107 sd=33 v=0.96 sd=0.13 cor=0.99

Q10 is Q at 10 Hz.

27.0.2 Program operation

The program first reads the parameter file, default codaq.par which must be in your current directory. It then reads the codaq.inp file with the events to analyze (also in current directory). The S-file names given here can, as shown in the examples above, be in the database or elsewhere, e.g. in your local directory. In the S-file, the name of the waveform file is given. If more than one waveform file is given, all files will be searched for the specified station and component. The program will first look in the current directory, and then in WAV and thereafter in the WAV database and other directories as given in the
SEISAN.DEF file in DAT. The program can therefore work without moving the data from the database, however you can also move both the S-files and waveform files to your local directory. Remember that the S-files must be updated in order to have origin time, since the program uses the origin time and P arrival times from the S-files.

Running the program:

Type codaq, the program asks about output:

0: Only q is calculated, enter ris 0 for default
1: Q is calculated and a plot on the tek screen is shown
2: Q  
   and at the same time hard copy plots are made.
3: Q is calculated and hard copy plots are made, but 
   no screen plot.

Parameter file, name codaq.par is default (return) 
   Just hit return if default file, otherwise give name.

File with event stations, codaq.inp is default (return) 
   Just hit return if default file, otherwise give name.

The program will now start to run. Alternatively, the program can be started with arguments on the prompt line:

```
codaq n parameter-file data-file
```

or alternatively when doing channel averages

```
codaq -c n parameter-file data-file
```

and no questions are asked. n is the choice 0 to 3 above.

If no plot is chosen, one line will appear on the screen for each station used and one for each frequency. The program will start a new page for each new event. If you are plotting on the screen, you will therefore have to hit return to get the next plot. The screen might not have been filled out if there are few data.

All questions will appear in the text window. At the end, a summary is given, which is the same as logged in the output file `codaq.out`.

The abbreviations are:
H: Focal depth
M: Magnitude
TP: P travel time
TC: Start time of coda window relative to origin time
F: Frequency
Q: Corresponding coda q, if 0 value is > 10000 or negative
S/N: Signal to noise ratio AV
Q: Average q
SD: Standard deviation for average
NT: Total number of q values at all frequencies
N: Number of q values at given frequency
q: Average of q values
1/q: q is calculated as 1/q averages, probably the best to use
f:1/q: Q values calculated using the relation derived from the 1/q averages
q = q0*f**v obtained with the average 1/q-values
cq0: Constant q0 obtained using the fixed user selected v
v: Constant v determined
cor: Correlation coefficient on determining q vs f
corr: Average correlation coefficients of individual codaq calculations
when fitting the envelope, both average and standard deviation is given

If a station is not present or no P is read, a message will be given. The program will search for the first P arrival time in the S-file. If several are present for the same station, it will use the first.

27.0.3 Output files

codaq.out

A file called codaq.out is generated. It contains a copy of the parameter file, one line for each event station combination accepted by the program (correlation and s/n ratio) and the average q values. The q values are averaged directly (indicated by q) and 1/q are averaged (indicated by 1/q). At the end are the fits to the $q = q0 * f * v$ relation. A summary of codaq.out is given in codaq1.out. This relation is calculated using the average Q-values for each frequency and each average is weighted by the number of observations used to calculate the average.

codaq.area

Output of codaq midpoints, codaq.area. For each accepted Q value, the midpoint between the station and the event together with the corresponding Q and frequency is saved in the file, see example below

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
<th>Lat</th>
<th>Long</th>
<th>Time</th>
<th>Channel</th>
<th>Day</th>
<th>Start</th>
<th>End</th>
<th>Q</th>
<th>f</th>
<th>cor</th>
<th>corr</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>117201030</td>
<td>34.58</td>
<td>112.67</td>
<td>1.0</td>
<td>BH Z</td>
<td>91.2</td>
<td>10.0</td>
<td>37.5</td>
<td>38.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>117201030</td>
<td>34.58</td>
<td>112.67</td>
<td>2.0</td>
<td>BH Z</td>
<td>223.9</td>
<td>10.0</td>
<td>37.5</td>
<td>38.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>117201030</td>
<td>34.58</td>
<td>112.67</td>
<td>4.0</td>
<td>BH Z</td>
<td>444.0</td>
<td>10.0</td>
<td>37.5</td>
<td>38.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>117201030</td>
<td>34.58</td>
<td>112.67</td>
<td>8.0</td>
<td>BH Z</td>
<td>833.4</td>
<td>10.0</td>
<td>37.5</td>
<td>38.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>117201030</td>
<td>34.58</td>
<td>112.67</td>
<td>16.0</td>
<td>BH Z</td>
<td>1324.1</td>
<td>10.0</td>
<td>37.5</td>
<td>38.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>117201030</td>
<td>34.58</td>
<td>112.67</td>
<td>1.0</td>
<td>BH N</td>
<td>132.2</td>
<td>10.0</td>
<td>37.5</td>
<td>38.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>117201030</td>
<td>34.58</td>
<td>112.67</td>
<td>2.0</td>
<td>BH N</td>
<td>255.3</td>
<td>10.0</td>
<td>37.5</td>
<td>38.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>117201030</td>
<td>34.58</td>
<td>112.67</td>
<td>4.0</td>
<td>BH N</td>
<td>374.2</td>
<td>10.0</td>
<td>37.5</td>
<td>38.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The content is: Event date, station code, component, lat-lon of midpoint, frequency, Q, sd in Q, depth, epicentral distance and hypocentral distance. This information can be used to plot the areal variation in Q and SEISAN provides one such program, CODAQ_AREA, see below. It can also be used for statistics with QSTAT, see below.

**codaq.index**

A file called `codaq.index` is created. This index file contains all the events accepted for calculating the codaq values and can therefore e.g. be used for making an epicenter map of events actually used (use collect with the index file)

**codaq1.out**

Output file `codaq1.out` contains the same output as `codaq.out` except there is no print out for each event. Additionally it has the average results for each station/component.

Example of `codaq.out`:

```
start in s-times 2.00
absolute start time (sec) 0.00
window length (sec) 15.00
spreading parameter 1.00
constant v in q = q0*f**v 1.00
minimum signal to noise ratio 5.00
noise window in front of signal and len 15.00 5.00
minimum correlation coefficient 0.50
maximum counts to use 50000
ASK SUE KMY EGD HYA
S Z S Z S Z S Z
\SEISMO\WAV\1996-06-07-1324-51S.TEST__009
\SEISMO\WAV\1996-06-07-1324-51S.TEST__009
\SEISMO\WAV\1996-06-07-1324-51S.TEST__009
1996 6 7132458 KMY tc 41.3 f 16.0 s/n 37.4 Q 1077 corr -0.55 rms 0.30
\SEISMO\WAV\1996-06-07-1324-51S.TEST__009
\SEISMO\WAV\1996-06-07-1324-51S.TEST__009
\SEISMO\WAV\1996-06-07-1324-51S.TEST__009
1996 6 625 33715 ASK tc 87.2 f 4.0 s/n121.9 Q 340 corr -0.60 rms 0.19
1996 6 625 33715 ASK tc 87.2 f 8.0 s/n 72.5 Q 551 corr -0.56 rms 0.28
\SEISMO\WAV\1996-06-07-1324-51S.TEST__009
1996 6 625 337 5 SUE tc 57.3 f 4.0 s/n 96.9 Q 193 corr -0.61 rms 0.34
\SEISMO\WAV\1996-06-07-1324-51S.TEST__009
1996 6 625 33730 KMY tc 143.5 f 8.0 s/n 13.6 Q 506 corr -0.61 rms 0.27
```
Above, the one line per q calculations is showing results from different stations. Only the traces selected (fulfilling selection criteria) are shown. The time indicated, is the start time in the waveform file for that particular station. In general, the start time for each channel of digital data would be different. If some data is missing, it is also show in the *codaq.out* file. Corr is the average correlation coefficient (with standard deviation) for the data selected for that frequency. The average lapse time is the average of the tc - values.

codaqxx.stat is

Q-values at each frequency xx is given in file codaq08.stat is for e.g. 8 Hz. The file can be used to plot the results with other programs.

codaq.summary

the files give a summary of the runs, e.g.

ntotal= 1096 q0= 68 sd= 3 v= 0.95 sd= 0.02 cor= 1.00

codaq.channel

The file gives the average values for each station, see example below:

<table>
<thead>
<tr>
<th>Station</th>
<th>n</th>
<th>q10</th>
<th>q0</th>
<th>sd</th>
<th>v</th>
<th>sd</th>
<th>cor</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LYN</td>
<td>21</td>
<td>622</td>
<td>64</td>
<td>4</td>
<td>0.99</td>
<td>0.03</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>YMG</td>
<td>88</td>
<td>570</td>
<td>68</td>
<td>6</td>
<td>0.92</td>
<td>0.04</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>DAX</td>
<td>16</td>
<td>701</td>
<td>78</td>
<td>7</td>
<td>0.95</td>
<td>0.04</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>TTY</td>
<td>38</td>
<td>668</td>
<td>81</td>
<td>9</td>
<td>0.92</td>
<td>0.06</td>
<td>0.99</td>
<td></td>
</tr>
<tr>
<td>XAX</td>
<td>40</td>
<td>605</td>
<td>67</td>
<td>10</td>
<td>0.95</td>
<td>0.07</td>
<td>0.99</td>
<td></td>
</tr>
<tr>
<td>LIF</td>
<td>50</td>
<td>719</td>
<td>57</td>
<td>3</td>
<td>1.10</td>
<td>0.03</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>HEYT</td>
<td>12</td>
<td>676</td>
<td>77</td>
<td>18</td>
<td>0.94</td>
<td>0.12</td>
<td>0.98</td>
<td></td>
</tr>
<tr>
<td>Station</td>
<td>n</td>
<td>q10</td>
<td>q0</td>
<td>sd</td>
<td>v</td>
<td>sd</td>
<td>cor</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>----</td>
<td>-----</td>
<td>-----</td>
<td>----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td>KEL</td>
<td>34</td>
<td>666</td>
<td>60</td>
<td>6</td>
<td>1.05</td>
<td>0.05</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>DOS</td>
<td>105</td>
<td>629</td>
<td>63</td>
<td>4</td>
<td>1.00</td>
<td>0.04</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>TAG</td>
<td>24</td>
<td>694</td>
<td>73</td>
<td>8</td>
<td>0.98</td>
<td>0.06</td>
<td>0.99</td>
<td></td>
</tr>
<tr>
<td>JIC</td>
<td>90</td>
<td>663</td>
<td>79</td>
<td>5</td>
<td>0.92</td>
<td>0.03</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>SZZ</td>
<td>72</td>
<td>617</td>
<td>80</td>
<td>4</td>
<td>0.89</td>
<td>0.03</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>HSH</td>
<td>114</td>
<td>511</td>
<td>78</td>
<td>2</td>
<td>0.82</td>
<td>0.01</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>HZH</td>
<td>16</td>
<td>698</td>
<td>75</td>
<td>5</td>
<td>0.97</td>
<td>0.03</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>JIN</td>
<td>10</td>
<td>706</td>
<td>74</td>
<td>6</td>
<td>0.98</td>
<td>0.04</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>GUJ</td>
<td>12</td>
<td>756</td>
<td>71</td>
<td>14</td>
<td>1.03</td>
<td>0.10</td>
<td>0.99</td>
<td></td>
</tr>
<tr>
<td>LIS</td>
<td>4</td>
<td>651</td>
<td>125</td>
<td>0</td>
<td>0.72</td>
<td>0.00</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>XIX</td>
<td>11</td>
<td>492</td>
<td>61</td>
<td>22</td>
<td>0.91</td>
<td>0.16</td>
<td>0.96</td>
<td></td>
</tr>
<tr>
<td>YUQ</td>
<td>52</td>
<td>598</td>
<td>62</td>
<td>5</td>
<td>0.98</td>
<td>0.04</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>YAY</td>
<td>81</td>
<td>621</td>
<td>66</td>
<td>5</td>
<td>0.98</td>
<td>0.04</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>HMA</td>
<td>22</td>
<td>638</td>
<td>76</td>
<td>3</td>
<td>0.92</td>
<td>0.02</td>
<td>1.00</td>
<td></td>
</tr>
</tbody>
</table>

Codaq.area.nor This file gives the midpoint coordinates in Nordic format so ready to plot with epimap and other mapping programs. Example below.

```
1999 210 1052 56.0 L 38.946 -9.168 3.0XXXX 1
1999 210 1052 56.0 L 38.946 -9.168 3.0XXXX 1
1999 210 1052 56.0 L 38.946 -9.168 3.0XXXX 1
1999 210 1052 56.0 L 39.092 -9.118 3.0XXXX 1
```

In the DAT directory, there is an example codaq.par and codaq.inp set up to run on PC assuming that SEISAN has been installed under seismo. If installed differently, edit the codaq.inp file to reflect the installation. For Unix testing, the codaq.inp MUST be edited to reflect the installation path or the file is regenerated using EEV as described above.

### 27.0.4 General recommendations and example of test data

Coda window should be at least 30 seconds, minimum correlation coefficient larger than 0.6. For comparing coda values in different regions, ALL processing parameters must be identical including lapse time. In [Havskov et al., 2016](#) the effect of the different parameters is investigated and a set of recommendations are given. The parameters found in the example file in DAT are the parameters used in [Havskov et al., 2016](#). In DAT there is also an input file codaq.inp using available data in the test data base. The input file has been created with program SELECT (option 18, station code: CODAQ, distance range 0 to 100 km). The start of the file is

```
1 /seismo/REA/TEST_/1990/12/13-1109-19L.S199012
SUE SUE SUE ODD1 ODD1 ODD1 ASK ASK ASK
```
which assumes that the test data base is installed under seismo. In a standard Windows installation, that would work. Under Linux it must be change to something like /home/seismo.... Alternatively create the file with SELECT.

Figure 27.1 gives an example of a codaq plot. There are no options for the codaq plots and the length of the window is always the first 200 secs from the original trace. If origin time or coda window is outside this 200-sec window and data is available, the program continues, but the coda window is not plotted on the figure.

27.1 Program CODAQ.AREA

CODAQ outputs the Q-values at midpoints between station and events. Using these values directly will often not show consistent results so some averring is needed. CODAQ.AREA reads this output and, at each frequency, average Q-values in user defined lat-lon bins. In each bin, provided there is enough data, the Q relation

\[ Q = Q_0 f^{q \alpha} \]

is also calculated. If little data is available, \( Q_0 \) can additionally be calculated for fixed \( q \alpha \) given by the user. Results are also made for \( f=10 \) Hz, \( Q_{10} \), since this might show the differences between areas more clearly. The values in each bin is listed in an output file which can be displayed to get an approximate idea of the areal Q-distribution. In addition, a contoured map is made with EPIMAP of \( Q_0 \) or \( Q_{10} \) and started from CODAQ.AREA. Optionally it can be deselected with parameter *Extrapolate*, see below.

Input files: It is assumed that codaq.area and codaq1.out exist so these names are hardwired. Codaq1.out is used to get the frequencies available for the data set.

The interactive options are: Min no for grid average, min no of freq for \( Q_0 \) calculation: Minimum number of values in a grid point for average to be calculated

Constant \( q \alpha \) in \( Q = Q_0 f^{q \alpha} \) relation, 0 means do not use: \( Q_0 \), can optional be calculated with a fixed \( q \alpha \). This can be useful when there is little data and it is expected that \( q \alpha \) does not vary a lot over a region.

Print numbers of observations for each \( Q_0 \): 1 else 0: When printing out the \( Q_0 \), it is also possible to print out the total number of observations used for calculating \( Q_0 \). This is all results for all frequencies.

Extrapolate \( Q_0 \) or \( Q_{10} \) for plot, 0=no, 1=yes, -1= do no plot contours: When making the contours, it is possible to extrapolate the \( Q_0 \) or \( Q_{10} \) values to bins with no data. This is done by averaging the values for the bins around. This process is done several times to fill the complete grid so in gridpoints more than 2 away from grids with data, all values will be constant.

Plot \( Q_0 \) (0) or \( Q_{10} \) (1): Plot \( Q_0 \) or \( Q_{10} \)
Figure 27.1: An example of a coda Q plot. On top is shown the original trace and below the filtered coda windows. Note that 15 secs of noise are shown in front of the selected filtered coda window. The first 5 secs of the noise shown is used for calculating the S/N ratio. On each filtered plot is given F: Center frequency, Q: Q-value, zero means no Q-value could be calculated, S/N: Signal to noise ratio.
Latitude range and grid size: It can be a fraction of a degree but Q0 does not have resolutions better than one degree.

Longitude range and grid size: ——————————–

The interactive options can be stored in a file called codaq_area.inp. If the file is present, input will be read from that file and no questions asked. An example of the file is:

Minimum number for grid average
3
Minimum number of frequencies for Q0 calculation
4
Constant qalpha in Q=Q0*f**qalpha relation, 0 means do not use
0.0
Print numbers of observations for each Q0 (1), else 0
0
Extrapolate q0 or q10 for plot, 0=not, 1=yes
1
Plot Q0 (0) or q10 (1)
0
Latitude range and delta
34 44 1
Longitude range and delta
-12,-2 1

The comments between the lines with numbers are just for information but there must be a line. This example file is also in DAT.

Example run with input from parameter file so no questions:

C:\>codaq_area
Frequencies 1.00 2.00 4.00 8.00 16.00
Min no for grid average, min no of freq for Q0 calculation 3 4
Constant in Q-relation, print number of observations 0.00000000
Extrapolate, plot Q0 0 0
Lat range and delta 34.0000000 44.0000000 1.00000000
Lon range and delta -14.0000000 -2.00000000 1.00000000
Number of q-data in input file 4581
Number of q-data inside grid 4581

Range of Q values to plot 58.4 310.7
Writing codaq_area\_epimap.inp
....
.... epimap output
....

File with area grid: codaq_area.out
File with grid points: codaq_grid.out
Example of the areal output file coda_area.out showing the lat-lon bins, note the midpoint of the bin is used:

<table>
<thead>
<tr>
<th>freq= 1.00000000</th>
<th>-13.5</th>
<th>-12.5</th>
<th>-11.5</th>
<th>-10.5</th>
<th>-9.5</th>
<th>-8.5</th>
<th>-7.5</th>
<th>-6.5</th>
<th>-5.5</th>
<th>-4.5</th>
<th>-3.5</th>
<th>-2.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>43.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>42.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>39.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>81</td>
<td>93</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>38.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>136</td>
<td>114</td>
<td>0</td>
<td>0</td>
<td>196</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>37.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>88</td>
<td>94</td>
</tr>
<tr>
<td>36.5</td>
<td>0</td>
<td>0</td>
<td>97</td>
<td>0</td>
<td>0</td>
<td>141</td>
<td>0</td>
<td>83</td>
<td>124</td>
<td>0</td>
<td>103</td>
<td>0</td>
</tr>
<tr>
<td>35.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>134</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>107</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>34.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>89</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>freq= 2.00000000</th>
<th>-13.5</th>
<th>-12.5</th>
<th>-11.5</th>
<th>-10.5</th>
<th>-9.5</th>
<th>-8.5</th>
<th>-7.5</th>
<th>-6.5</th>
<th>-5.5</th>
<th>-4.5</th>
<th>-3.5</th>
<th>-2.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>43.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>256</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>42.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>323</td>
<td>0</td>
<td>227</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>394</td>
<td>150</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>112</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>39.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>221</td>
<td>227</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>38.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>256</td>
<td>338</td>
<td>303</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>37.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>245</td>
<td>273</td>
<td>0</td>
<td>0</td>
<td>126</td>
<td>125</td>
<td>0</td>
</tr>
<tr>
<td>36.5</td>
<td>0</td>
<td>219</td>
<td>141</td>
<td>0</td>
<td>0</td>
<td>156</td>
<td>0</td>
<td>133</td>
<td>155</td>
<td>0</td>
<td>135</td>
<td>0</td>
</tr>
<tr>
<td>35.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>155</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>155</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>34.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>128</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>freq= 4.00000000</th>
<th>-13.5</th>
<th>-12.5</th>
<th>-11.5</th>
<th>-10.5</th>
<th>-9.5</th>
<th>-8.5</th>
<th>-7.5</th>
<th>-6.5</th>
<th>-5.5</th>
<th>-4.5</th>
<th>-3.5</th>
<th>-2.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>43.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>487</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>42.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>619</td>
<td>0</td>
<td>411</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>640</td>
<td>485</td>
<td>565</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>502</td>
<td>546</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>39.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>556</td>
<td>531</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>38.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>503</td>
<td>560</td>
<td>540</td>
<td>0</td>
<td>0</td>
<td>623</td>
<td>0</td>
</tr>
<tr>
<td>37.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>319</td>
<td>500</td>
<td>505</td>
<td>0</td>
<td>265</td>
<td>235</td>
<td>207</td>
<td>0</td>
</tr>
<tr>
<td>36.5</td>
<td>0</td>
<td>287</td>
<td>210</td>
<td>384</td>
<td>280</td>
<td>303</td>
<td>287</td>
<td>196</td>
<td>211</td>
<td>0</td>
<td>204</td>
<td>0</td>
</tr>
<tr>
<td>35.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>273</td>
<td>0</td>
<td>0</td>
<td>207</td>
<td>0</td>
<td>233</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>34.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>209</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>freq= 8.00000000</th>
<th>-13.5</th>
<th>-12.5</th>
<th>-11.5</th>
<th>-10.5</th>
<th>-9.5</th>
<th>-8.5</th>
<th>-7.5</th>
<th>-6.5</th>
<th>-5.5</th>
<th>-4.5</th>
<th>-3.5</th>
<th>-2.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>43.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>790</td>
<td>803</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>42.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>921</td>
<td>1121</td>
<td>0</td>
<td>712</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>859</td>
<td>896</td>
<td>808</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>40.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>732</td>
<td>797</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### 27.1. PROGRAM CODAQ_AREA

<table>
<thead>
<tr>
<th>freq= 16.0000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>-13.5 -12.5 -11.5 -10.5 -9.5 -8.5 -7.5 -6.5 -5.5 -4.5 -3.5 -2.5</td>
</tr>
<tr>
<td>43.5 0 0 0 0 0 1238 1306 1342 0 0 0 0</td>
</tr>
<tr>
<td>42.5 0 0 0 0 0 1362 1574 0 1216 0 0 0 0</td>
</tr>
<tr>
<td>41.5 0 0 0 0 0 0 0 0 1499 1302 0 0 0 0</td>
</tr>
<tr>
<td>40.5 0 0 0 0 0 0 0 0 1451 0 0 0 0 0</td>
</tr>
<tr>
<td>39.5 0 0 0 0 0 0 0 0 1558 1376 0 0 0 0 0</td>
</tr>
<tr>
<td>38.5 0 0 0 0 0 0 0 0 1534 1375 1295 1442 0 1537 0 0</td>
</tr>
<tr>
<td>37.5 0 0 0 0 0 1413 1306 1366 1418 0 970 805 0</td>
</tr>
<tr>
<td>36.5 0 0 0 0 0 1007 1427 0 0 0 0 0 0</td>
</tr>
<tr>
<td>35.5 0 0 0 0 0 0 0 0 0 1188 0 0 0 0 0</td>
</tr>
<tr>
<td>34.5 0 0 0 0 0 0 0 0 0 0 989 0 0 0 0</td>
</tr>
</tbody>
</table>

### Q0

<table>
<thead>
<tr>
<th>-13.5 -12.5 -11.5 -10.5 -9.5 -8.5 -7.5 -6.5 -5.5 -4.5 -3.5 -2.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>43.5 0 0 0 0 0 160 0 0 0 0 0</td>
</tr>
<tr>
<td>42.5 0 0 0 0 0 226 0 0 133 0 0 0 0</td>
</tr>
<tr>
<td>41.5 0 0 0 0 0 256 129 0 0 0 0 0</td>
</tr>
<tr>
<td>40.5 0 0 0 0 0 58 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>39.5 0 0 0 0 0 103 126 0 0 0 0 0 0</td>
</tr>
<tr>
<td>38.5 0 0 0 0 0 142 162 193 0 196 266 0 0</td>
</tr>
<tr>
<td>37.5 0 0 0 0 0 69 169 166 310 0 85 77 0</td>
</tr>
<tr>
<td>36.5 0 114 83 0 0 87 0 0 0 0 0 0</td>
</tr>
<tr>
<td>35.5 0 0 0 0 0 0 0 0 93 0 0 0</td>
</tr>
<tr>
<td>34.5 0 0 0 0 0 0 0 0 0 72 0 0 0</td>
</tr>
</tbody>
</table>

### Q10

<table>
<thead>
<tr>
<th>-13.5 -12.5 -11.5 -10.5 -9.5 -8.5 -7.5 -6.5 -5.5 -4.5 -3.5 -2.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>43.5 0 0 0 0 0 0 0 0 923 0 0 0 0 0</td>
</tr>
<tr>
<td>42.5 0 0 0 0 0 1032 0 0 841 0 0 0 0 0</td>
</tr>
<tr>
<td>41.5 0 0 0 0 0 1115 1005 0 0 0 0 0 0</td>
</tr>
<tr>
<td>40.5 0 0 0 0 0 1468 0 0 0 0 0 0 0</td>
</tr>
<tr>
<td>39.5 0 0 0 0 0 1061 965 0 0 0 0 0 0</td>
</tr>
<tr>
<td>38.5 0 0 0 0 0 1035 1034 962 0 1193 1150 0 0</td>
</tr>
<tr>
<td>37.5 0 0 0 0 0 848 933 1006 1096 0 522 457 0</td>
</tr>
<tr>
<td>36.5 0 559 440 0 0 857 0 0 0 0 0 0 417 0</td>
</tr>
<tr>
<td>35.5 0 0 0 0 0 0 0 0 561 0 0 0</td>
</tr>
<tr>
<td>34.5 0 0 0 0 0 0 0 0 0 590 0 0 0</td>
</tr>
</tbody>
</table>

### Qalpha

<table>
<thead>
<tr>
<th>-13.5 -12.5 -11.5 -10.5 -9.5 -8.5 -7.5 -6.5 -5.5 -4.5 -3.5 -2.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>43.5 0.00 0.00 0.00 0.00 0.00 0.00 0.76 0.00 0.00 0.00 0.00 0.00</td>
</tr>
<tr>
<td>42.5 0.00 0.00 0.00 0.00 0.66 0.00 0.00 0.80 0.00 0.00 0.00 0.00 0.00</td>
</tr>
<tr>
<td>41.5 0.00 0.00 0.00 0.00 0.00 0.64 0.89 0.00 0.00 0.00 0.00 0.00 0.00</td>
</tr>
<tr>
<td>40.5 0.00 0.00 0.00 0.00 0.00 1.40 0.00 0.00 0.00 0.00 0.00 0.00</td>
</tr>
<tr>
<td>39.5 0.00 0.00 0.00 0.00 1.01 0.88 0.00 0.00 0.00 0.00 0.00 0.00 0.00</td>
</tr>
</tbody>
</table>
Output file codaq_grid.out contains details of the averages in each bin see part of file below:

<table>
<thead>
<tr>
<th>freq</th>
<th>1.00000000</th>
<th>34.500</th>
<th>-13.500</th>
<th>0.0</th>
<th>0.0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>34.500</td>
<td>-12.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34.500</td>
<td>-11.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34.500</td>
<td>-10.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34.500</td>
<td>-9.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34.500</td>
<td>-8.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34.500</td>
<td>-7.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34.500</td>
<td>-6.500</td>
<td>0.0</td>
<td>0.0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34.500</td>
<td>-5.500</td>
<td>89.5</td>
<td>21.1</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34.500</td>
<td>-4.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34.500</td>
<td>-3.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>34.500</td>
<td>-2.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35.500</td>
<td>-13.500</td>
<td>134.9</td>
<td>13.7</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35.500</td>
<td>-12.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35.500</td>
<td>-11.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35.500</td>
<td>-10.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35.500</td>
<td>-9.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35.500</td>
<td>-8.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35.500</td>
<td>-7.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35.500</td>
<td>-6.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35.500</td>
<td>-5.500</td>
<td>107.8</td>
<td>15.8</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35.500</td>
<td>-4.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35.500</td>
<td>-3.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>35.500</td>
<td>-2.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.500</td>
<td>-13.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.500</td>
<td>-12.500</td>
<td>0.0</td>
<td>0.0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.500</td>
<td>-11.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.500</td>
<td>-10.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.500</td>
<td>-9.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.500</td>
<td>-8.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.500</td>
<td>-7.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.500</td>
<td>-6.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.500</td>
<td>-5.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.500</td>
<td>-4.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.500</td>
<td>-3.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>36.500</td>
<td>-2.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>37.500</td>
<td>-13.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>x</td>
<td>y</td>
<td>x1</td>
<td>y1</td>
<td>x2</td>
<td>y2</td>
<td>x3</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>37.50</td>
<td>-12.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.50</td>
<td>-11.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.50</td>
<td>-10.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.50</td>
<td>-9.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.50</td>
<td>-8.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.50</td>
<td>-7.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.50</td>
<td>-6.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.50</td>
<td>-5.50</td>
<td>0.0</td>
<td>0.0</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.50</td>
<td>-4.50</td>
<td>88.6</td>
<td>13.9</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.50</td>
<td>-3.50</td>
<td>94.8</td>
<td>19.9</td>
<td>21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.50</td>
<td>-2.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.50</td>
<td>-13.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.50</td>
<td>-12.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.50</td>
<td>-11.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.50</td>
<td>-10.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.50</td>
<td>-9.50</td>
<td>136.3</td>
<td>53.6</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.50</td>
<td>-8.50</td>
<td>114.3</td>
<td>40.5</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.50</td>
<td>-7.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.50</td>
<td>-6.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.50</td>
<td>-5.50</td>
<td>196.6</td>
<td>17.3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.50</td>
<td>-4.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.50</td>
<td>-3.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.50</td>
<td>-2.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.50</td>
<td>-13.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.50</td>
<td>-12.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.50</td>
<td>-11.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.50</td>
<td>-10.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.50</td>
<td>-9.50</td>
<td>81.2</td>
<td>17.9</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.50</td>
<td>-8.50</td>
<td>93.5</td>
<td>30.9</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.50</td>
<td>-7.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.50</td>
<td>-6.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.50</td>
<td>-5.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.50</td>
<td>-4.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.50</td>
<td>-3.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.50</td>
<td>-2.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40.50</td>
<td>-13.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40.50</td>
<td>-12.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40.50</td>
<td>-11.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40.50</td>
<td>-10.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40.50</td>
<td>-9.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40.50</td>
<td>-8.50</td>
<td>0.0</td>
<td>0.0</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40.50</td>
<td>-7.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40.50</td>
<td>-6.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40.50</td>
<td>-5.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40.50</td>
<td>-4.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40.50</td>
<td>-3.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40.50</td>
<td>-2.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41.50</td>
<td>-13.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41.50</td>
<td>-12.50</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
41.500  -11.500  0.0  0.0  0
41.500  -10.500  0.0  0.0  0
41.500   -9.500  0.0  0.0  0
41.500   -8.500  0.0  0.0  0
41.500   -7.500  0.0  0.0  1
41.500   -6.500  0.0  0.0  0
41.500   -5.500  0.0  0.0  0
41.500   -4.500  0.0  0.0  0
41.500   -3.500  0.0  0.0  0
42.500  -13.500  0.0  0.0  0
42.500  -12.500  0.0  0.0  0
42.500  -11.500  0.0  0.0  0
42.500  -10.500  0.0  0.0  0
42.500   -9.500  0.0  0.0  0
42.500   -8.500  0.0  0.0  0
42.500   -7.500  0.0  0.0  0
42.500   -6.500  0.0  0.0  0
42.500   -5.500  0.0  0.0  0
42.500   -4.500  0.0  0.0  0
42.500   -3.500  0.0  0.0  0
42.500   -2.500  0.0  0.0  0
43.500  -13.500  0.0  0.0  0
43.500  -12.500  0.0  0.0  0
43.500  -11.500  0.0  0.0  0
43.500  -10.500  0.0  0.0  0
43.500   -9.500  0.0  0.0  0
43.500   -8.500  0.0  0.0  0
43.500   -7.500  0.0  0.0  0
43.500   -6.500  0.0  0.0  0
43.500   -5.500  0.0  0.0  0
43.500   -4.500  0.0  0.0  0
43.500   -3.500  0.0  0.0  0
43.500   -2.500  0.0  0.0  0
34.500  -13.500  0.0  0.0  0
34.500  -12.500  0.0  0.0  0
34.500  -11.500  0.0  0.0  0
34.500  -10.500  0.0  0.0  0
34.500   -9.500  0.0  0.0  0
34.500   -8.500  0.0  0.0  0
34.500   -7.500  0.0  0.0  0
34.500   -6.500  0.0  0.0  0
34.500   -5.500  129.0  23.5  114
34.500   -4.500  0.0  0.0  0
34.500   -3.500  0.0  0.0  0
34.500   -2.500  0.0  0.0  0
35.500  -13.500  0.0  0.0  0
35.500  -12.500  0.0  0.0  0
27.2. **PROGRAM CODAQ_POLYGON**

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>35.500</td>
<td>-11.500</td>
<td>0.0</td>
<td>0.0</td>
<td>1</td>
</tr>
<tr>
<td>35.500</td>
<td>-10.500</td>
<td>155.4</td>
<td>26.0</td>
<td>23</td>
</tr>
<tr>
<td>35.500</td>
<td>-9.500</td>
<td>0.0</td>
<td>0.0</td>
<td>1</td>
</tr>
<tr>
<td>35.500</td>
<td>-8.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>35.500</td>
<td>-7.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>35.500</td>
<td>-6.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>35.500</td>
<td>-5.500</td>
<td>155.6</td>
<td>13.4</td>
<td>11</td>
</tr>
<tr>
<td>35.500</td>
<td>-4.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>35.500</td>
<td>-3.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>35.500</td>
<td>-2.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>36.500</td>
<td>-13.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
<tr>
<td>36.500</td>
<td>-12.500</td>
<td>219.2</td>
<td>106.5</td>
<td>5</td>
</tr>
<tr>
<td>36.500</td>
<td>-11.500</td>
<td>141.4</td>
<td>30.3</td>
<td>7</td>
</tr>
<tr>
<td>36.500</td>
<td>-10.500</td>
<td>0.0</td>
<td>0.0</td>
<td>0</td>
</tr>
</tbody>
</table>

The output is: Bin midpoint, average Q, standard deviation in average and number of points in bin.

Finally a plot is made with contours of Q, see Figure 27.2.

The map with the contours is produced automatically from the input. The projection is Mercator and the land contour is WORLD.MAP in DAT. Another map file can be used by setting parameter EPIMAP_MAP_FILE in SEISAN.DEF. The commands for running EPIMAP are in output file codaq_area_epimap.inp, so this file can also be modified to change map projection etc (see EPIMAP) and the plot regenerated by giving command

epimap codaq_area_epimap.inp

### 27.2 Program CODAQ_POLYGON

Using EPIMAP, it is possible to select out events in a polygon when plotting the codaQ midpoints (or other epicentres). It is then possible to calculate the average Q in that area using the program CODAQ_POLYGON. The input polygon coordinates are assumed to be present in file epimap.cor, the one created by EPIMAP when selecting in a polygon. The codaQ values in the midpoints are supposed to be present in file codaq.area and no questions are asked. File codaq.out is also used as input and must be present. A depth range can be specified. The data in the polygon file is written out in a file POLYGON.MAP so it can be plotted again with EPIMAP. The POLYGON.MAP file is appended so if CODAQ_AREA is used several times with different polygon areas, they will all be stored in POLYGON.MAP read for plotting with EPIMAP. It is up to the user to ensure the POLYGON.MAP is empty if a new series of polygons are stored.

The section of the codaq.area corresponding to the polygon area is output in codaq_polygon.area and it can be used with QSTAT for statistics of the data in the polygon area.

Example run

codaq_polygon

c:\jens\Q-port\ALL>codaq_polygon
Figure 27.2: Map with contours. The 'epicenter' symbols show gridpoints with data and the size of the symbols show the total number of Q-results for that gridpoint, the number are shown to the left (not shown in real plot, have been put in manually). The color of the symbols indicate Q-values where the range of Q-values have been divided by 3 so blue is highest Q, green is average and red the smallest.
27.2. PROGRAM CODAQ\_POLYGON

Frequencies 1.00 2.00 4.00 8.00 16.00

Depth range to use, enter to use all

Number of points defining polygon: 7

f= 1.00000000

f= 2.00000000

216. 269. 308.

f= 4.00000000


f= 8.00000000

1366. 878. 968. 1295. 803. 999. 979. 967. 1155. 1080.

Number of Q-midpoints within polygon: 140

f= 2.0 av= 259.0 sd= 46.9 n= 3

f= 4.0 av= 628.9 sd= 119.7 n= 21

f= 8.0 av= 1035.7 sd= 163.2 n= 56

f= 16.0 av= 1531.8 sd= 167.4 n= 60

q0= 172.7 sd 38.8 qalpha= 0.81 sd 0.10 q10= 1107.8 sd= 182.1 corr= 0.98 rms= 0.00 nf= 4

Use all data, no weight

q0= 166.0 sd 36.8 qalpha= 0.84 sd 0.12 q10= 1151.4 sd= 188.4 corr= 0.98 rms= 0.00 nf= 4

Append to POLYGON.MAP (n/y=default)

Accumulated polygon coordinates in POLYGON.MAP

Area file selected is codaq\_polygon.area

For each frequency, all the Q-values are printed so it is possible to get an idea of the variation of the values. The averages are calculated for each frequency if 2 or more values are available. The Q-realition
calculated if averages are available for at least 3 frequencies. Finally, a Q-relation is calculated if also one value is available, but then there is no weighting with the standard deviations of the averages.

How to calculate coda Q in polygon areas

Calculate coda Q in large area of interest.

indexCoda Q in a polygon Plot all the midpoints between station end events with EPIMAP using command map codaq.area.nor. The input epicentre file is codaq.area.nor which is output from CODAQ.

Do the following for each polygon area:

With EPIMAP, use polygon function to select an area. Terminate EPIMAP.

Run codaq.polygon to calculate coda Q for that area, the polygon outline is stored in file POLYGON.MAP for later plotting.

After e.g. 5 runs, the 5 areas are stored in POLYGON.MAP and the Q-relations have been printed out.

### 27.3 Program QSTAT

This program also use the codaq.area to make statistics etc. After running CODAQ, codaq.area contains all individule coda Q determinations. The program first finds all the stations which has contributed to the calculations and makes an output file qstat.stat with station codes and station coordinates (require STATION0.HYP) so a map can be made with the ‘good’ stations. If a station is not found in STATION0.HYP, the lat-lon will be 0.0 0.0 in output file. Then a plot is made with Q as a function of distance for each frequency and a histogram is made with the the distribution of Q values for each frequency. The latter is intended to be illustrate the spread in the Q values and is useful when comparing Q from different areas. Note that the number of values for each frequency are not absolute, but scaled to fit the plot. The number of values for each frequency is indicated on the plot.

### 27.4 Program AVQ, average Q-relations

Q as a function of frequency is usually described as

\[ q = q_0 * f^v \]

If several such relations are to be averaged, it is not just a question of averaging the parameters. In program AVQ, the averaging is done in the following way:

- For each relation, 1/Q is calculated at the frequencies 1, 2, 4, 8 and 16 Hz. -At each frequency, average 1/Q is calculated using the number of observations in the original determination of Q for a particular relation as weight. -A new least squares determination of \( v \) in \( q_0 \) is made with the Q-values.

The program uses an input file with \( q_0, v \) and number of observations, one relation (free format) per line. An example of a run is seen below:

```bash
C:\seismo\wor>avq
File name, enter for automag_grid.out
input.txt
```
Figure 27.3: The top plot shows Coda Q as a function of epicentral distance and the bottom plot shows the distribution of individual Q-values for each frequency. $n$ is the number of Q-values for each frequency.
Figure 27.4: Figure AVQ: The figure shows the Q-relations to be averaged and the average Q-relation (red).

| Q0, alpha, n | 100.000000 | 0.500000000 | 100 |
| Q0, alpha, n | 150.000000 | 0.300000012 | 50  |
| Q0, alpha, n | 200.000000 | 0.200000003 | 10  |
| Q0, alpha, n | 170.000000 | 0.400000006 | 22  |
| Q0, alpha, n | 250.000000 | 0.150000006 | 5   |
| Q0, alpha, n | 80.0000000 | 0.800000012 | 10  |

Number of curves to average: 6
Running average over how many, enter for average of all?

Q0, alpha, corr 119.819160 0.437137932 0.999868274

Output of plot in avq.eps

and the plot seen in figure 27.4 comes up.

The program also has a special input to be used with AUTOMAG, which can output Q-relations found by grid search, see AUTOMAG for more details. These relations can be averaged over a number of
relations. However, here that option is not used.
Chapter 28

Merge events in SEISAN and compare catalogs

The programs described in this chapter are used for merging events in SEISAN as well as for comparing two catalogs.

28.1 Merge events near in time ASSOCI

The program will check if two events are close together in time and merge the events if requested. This is partly an alternative to use append in EEV. The program asks for maximum time difference between events to associate. The user will then be asked if events should be physically associated or not. The program is useful when merging a large number of events. The program has two alternatives for merging:

1. Merge events in same data base: One event is compared to the next event in the same data base. If they are close enough in time, the two events are merged and the program moves on to the next event. If 3 events are close in time, only the 2 first are merged. In order to also merge the third, the program has to be run again.

2. Merge events from a file into the data base: This option makes it possible to merge from another data base (use SELECT or COLLECT to create a file) without first completely mixing the two. The event from the file will be merged with as many files from the data base as fit the time difference criteria. So e.g. 2 events from the data base can both get the same event from the file included. At the end of the run, two files are output (associ_rest.out associ_merg.out) with events which were not merged and merged respectively. These can then be put into another data base with split, if desired. This function can also be used to separate the input file in two files.

Note: When merging within one data base, the first event will get the next one merged into it. If merging from file into a data base, the event in the data base will by default always be the first and keep the main header. This thus a safe method when you want to keep the main header unchanged in the data base. Optionally you can decide to always put the header from the file first.

Putting new data in a data base where locations are available in one place and waveform files have to be merged in:
CHAPTER 28. MERGE EVENTS IN SEISAN AND COMPARE CATALOGS

Use AUTOREG to make a data base with s-files with only waveform file names. This is done by using the option to copy or move waveform files to the WAV structure and the s-files will be made in a data base of the same name.

Use ASSOCI to merge in the s-files with locations and phases from a file and use the option to put the headers from the file first.

If the event files are put into the data base first, then the s-files from AUTOREG will also go into the data base (with option to move or copy waveform files the WAV data base) and ASSOCE will have to merge files which all are in the data base. Then there is no control which header comes first.

28.2 Merge events near in time and delete duplicate phases, MERGE_S

This program works in a similar way as ASSOCI, however with a better control of the merging.

The program can merge two files or a data base and a file. The two inputs are main input and merge input. The main input can be a data base or a file while merge input only can be a file. The merge input file is merged into the main input and the merged file written to an output file. None of the inputs are changed.

Events from the merge file is compared to events in the main input and merged if within user given time difference.

Duplicate phases are deleted if from same station with a time difference of less than 0.1 s. There is no check for component since some input sources, like from ISC, have no component. The user can specify if the phase from main input or merge input will be deleted. Normally the whole phase name is compared, but there is an option to use only the first letter. This can be used if two agencies use different upper and lower case like PN or Pn so only the first letter can be compared. However, this does not apply to SPEC phases since only used in SEISAN and we do not want to confuse S with SPEC.

For spectral phases, also component is used since that is always given. The spectral averages are taken from the main input. If no averages in main but in merge, they are taken from the merge input. So spectral averages might not be a correct representation of SPEC lines are merged, but that will be fixed with next update.

The first header and ID line will come from the main input file. The Id line from the merge file is converted to a comment line.

The locality line from main is kept and if there is a locality line in merge file, it is renamed from LOCALITY: to MER-LOCA:

Both inputs must be in chronological order, if not the program stops. Input from a data base (or using e.g. a collect from a data base) might therefore require can update with new ID lines.

Input can be old or new format but output is the one set up by the SEISAN.DEF.

There is no check for distance id.

All events from main input is written out again and events from merge file that does not match in time can optionally be put in main output file or in a separate file.

Example run:
28.3. **MERGE EVENTS NEAR IN TIME, DISTANCE, DEPTH AND MAGNITUDE ASSO**

The purpose of ASSO is to merge events which potentially are the same. In contrast to the ASSOCI program, ASSO will use more parameters to determine if events are identical: Time difference, magnitude...
CHAPTER 28. MERGE EVENTS IN SEISAN AND COMPARE CATALOGS

difference, epicentral distance and depth distance. The ASSOCI program only uses time difference. The parameters can be magnitude dependent. The program requires an input parameter file asso.def which can be placed in working directory or DAT. An example is given in DAT and seen below.

If the event has no location or magnitude, it will not be considered as a possible duplicate event and will be included as it is.

The magnitude used are selected among all prime magnitudes in the file (the 6 magnitudes on main header line, can be 2 lines if more than 3 magnitudes). The magnitudes can be given an order of priority in SEISAN.DEF:

<table>
<thead>
<tr>
<th>KEYWORD............</th>
<th>Comments.............</th>
<th>Par 1.....</th>
<th>Par 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAGNITUDE_ORDER LBER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAGNITUDE_ORDER LNAO</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In this example, LBER (local magnitude calculated by BER) is first chosen, if not there, LNAO and if not there either, the first magnitude found in file. If no order is given in SEISAN.DEF, the magnitude used will be the first found for the event, irrespective of type or agency. It is possible to leave either magnitude or agency blank in which case the blank represents a wildcard.

Potentially all events can be a main shock so all combinations, within the time limit, will be searched. However, one an event has been selected to be a duplicate, it cannot be used again for either main shock or duplicate.

Input:
A SEISAN data base. The data base is not modified by ASSO.

Output:
All duplicate events will be merged with the Main event. The largest event will be listed first. The output file with merged events is asso.out. Optionally, a debug output can be made, either on screen or in a file (see asso.def). An example is seen below

Main : 1995 1120 4 1 59.1 60.1 5.4 1.5 2.0 LTES
Main : 1996 3 1955 35.7 47.8 153.2 0.1 4.8 STES
Asso : 1996 6 3 1955 37.9 47.8 153.2 0.1 4.8 STES M0.7 T 46 2 D 38 0
Asso : 1996 6 3 1955 57.9 47.8 153.2 0.1 4.8 STES M0.7 T 46 22 D 38 0
Main : 1996 6 6 648 30.4 62.6 5.1 15.0 2.9 LTES
Main : 1996 6 6 648 30.4 62.6 5.1 15.0 2.9 LTES M0.5 T 9 0 D 21 0
Asso : 1996 6 6 648 30.4 62.6 5.1 15.0 2.9 LTES M0.5 T 9 0 D 21 0
Main : 1996 6 7 1325 29.2 59.8 5.1 12.0 1.9 LTES
Main : 1996 6 7 1325 29.2 59.8 5.1 12.0 1.9 LTES
Main : 1996 623 117 57.8 51.7 160.0 30.7 3.5 sTES
Main : 1996 623 117 57.8 51.6 159.6 33.0 5.8 bTES
Main : 1996 625 337 31.7 61.6 3.3 34.8 3.5 CTES
Asso : 1996 625 337 31.0 61.7 3.3 15.0 3.3 LTES M0.6 T 22 0 D 26 10
Main : 1996 7 5 220 45.9 61.3 4.8 0.0 2.7 CTES

Each potential main (Main) event is listed. To be considered potential main event, it must have both location and magnitude. If a duplicate event is found (Asso), it is shown with the parameters: M: The interpolated maximum magnitude difference used, T: The interpolated maximum time difference (sec)
used and the actual time difference (sec). D: The interpolated maximum distance (km) used and the actual distance. In the above example, the lower magnitude difference limit for the first associated shown is 0.7, the limit in distance is 38 km and in time difference 46 sec. The Main event is listed first, then all associated events so the list as shown above might not be in chronological order if there is an event within the accepted time range which do not fulfill the other criteria. In the above example, the Main event (1996 6 3 1955 35.7) has been merged with the 2 following events shown. But in between there is event (1996 6 3 1955 39.4 47.9 153.2 0.1 5.6 bPDE) which has a too large magnitude to be merged with the Main event, so this will then appear out of order and will be a potential new Main event. Once an event has been merged with a Main event, it will not be merged with another Main event.

Merging across day boundaries: This usually does not create problems. If the first event is before midnight and the second event after midnight, phase readings from the second event might have 24 h added if the readings are from the second day. However in some cases, ASSO will select the first event after midnight and the second event before midnight due to the magnitude size. In this case the header cannot refer to the earlier phase times since they cannot be negative and the phases for the second event are not merged. A message will be given.

Example of a asso.def file

This file is parameter file for asso. The file can be in working directory or DAT.
Only the lines with recognized keywords and a non blank field under Par 1 will be read. The comments have no importance. Lines with keywords MAGS MDIF DIST TIME must come grouped together in increasing magnitude order. Parameters for these lines are:
Reference magnitude, minimum magnitude difference of corresponding event to merge, distance (km) and time (sec).
All number must be real numbers. Columns Par 1-Par 4 start in columns 41, 51, 61 and 71 respectively.
All keywords in capital letters.
"Hypocentral" distance is calculated assuming main shock is at depth h1 and associated event at depth h2, epicentral distance at dist so hypocentral distance is = sqrt(dist*dist+(h2-h1)*(h2-h1)).

If an event has no depth (field is blank), the depth is set to 20 km.

If event is larger than largest event in table, largest event is used.
If event is smaller than smallest event in table, event is not used as main event.
Values used are interpolated.
The values shown in debug output are the interpolated values.
The MAGS MDIF DIST TIME are mag of main shock, corresponding maximum mag difference of associated event, maximum distance (km) and time (sec) differences.

The MAX DEPT DIFFERENCE will consider only events with a depth difference less than MAX DEPT DIFFERENCE as possible associated events.

The largest of the associated events is written out first.
KEYWORD............Comments.............Par 1......Par 2......Par 3......Par4......
HYPOCENTRAL DIST 1.0 yes, 0.0 no 0.0
MAX DEPTH DIFF Difference in depth 50.0
DEBUG OUT 0.0 none, 1.0 screen 2.0 file 1.0
# MAG MAG DIFF DIST DIFF TIME DIFF
MAGS MDIF DIST TIME Mag. dep. par. 2.5 0.5 19.5 6.0
MAGS MDIF DIST TIME Mag. dep. par. 3.0 0.5 22.5 10.0
MAGS MDIF DIST TIME Mag. dep. par. 3.5 0.6 26.0 22.0
MAGS MDIF DIST TIME Mag. dep. par. 4.0 0.6 30.0 30.0
MAGS MDIF DIST TIME Mag. dep. par. 4.5 0.7 35.0 40.0
MAGS MDIF DIST TIME Mag. dep. par. 5.0 0.7 40.0 50.0
MAGS MDIF DIST TIME Mag. dep. par. 5.5 0.8 47.0 60.0
MAGS MDIF DIST TIME Mag. dep. par. 6.0 0.8 54.0 70.0
MAGS MDIF DIST TIME Mag. dep. par. 6.5 0.9 61.0 75.0
MAGS MDIF DIST TIME Mag. dep. par. 7.0 0.9 70.0 80.0
MAGS MDIF DIST TIME Mag. dep. par. 7.5 1.0 81.0 90.0
MAGS MDIF DIST TIME Mag. dep. par. 8.0 1.0 94.0 90.0
MAGS MDIF DIST TIME Mag. dep. par. 8.5 1.0 110.0 100.0

28.4 Catalogue merging and comparison with merge_seisan.pl

This program was provided by Frederik Tilmann, (tilmann at gfz-potsdam.de) (c) 2002-2011.

merge_seisan.pl is a 'swiss-army knife' command line tool for catalogue merging and comparison. Unlike ASSO and ASSOCIATE, it operates always on two catalogues (either a catalogue file or extracted from the database). Its uses are

- Event association, for example search for events present in both catalogues and display them side by side (or to select only events exclusive to one of the catalogues)
- Merging information in two catalogues, for example to create a catalogue with Mw and focal mechanisms taken from one catalog (e.g. globalCMT), and hypocenter from another catalogue (e.g. one based on local network relocation).
- Comparison, i.e. generate output suitable for making histograms or map view comparisons of the locations in two catalogues, or calculate bias vectors between two catalogues.

Usage: merge_seisan.pl [options] cat1 cat2 > out.nor

Usage: merge_seisan.pl [options] cat1 cat2 > out.nor

Reads two CAT Files or Sfiles in nordic format and associates events, merging the information and respective files. Merged events have information from both files interspersed. Both files must be sorted. Also each event of the first set can only be associated with at most one event of the second set, and vice versa.
cat1 and cat2 can be one of the following:

- `<file.nor>` The filename of a file in Nordic format
- `DBASE` If cat1 is three to five letters long and a file of the same name does not exist the corresponding catalogues is created on the fly from the S-files in the respective database. If the code `DBASE` or `D` is used, use the current database

- read from standard in (only one `-` is allowed, of course)

Options:

Method of association:

- `-I` associate events by ID (default)
- `-T=tol` associate events within (tol) s of each other
- `-D=<dist>[h]` Maximum epicentral distance in km. If h is appended then the hypocentral distance is used instead.

- `-D` can be combined with `-T` but cannot be used on its own (both conditions need to be fulfilled).
- It also cannot be used with `-I` option

Output control

- `-S=I,-i` Only output associated events (Intersection)
- `-S=U,-u` Output all events (Union)
- `-S=A` Output only events in A (only makes sense when combined with `-b` or `-B` option)
- `-S=B` Output only events in B
- `-S=~A` Output only events not in A (i.e. only in B, only makes sense with `-B` option)
- `-S=~B` Output only events not in B (i.e. only in A, only makes sense with `-A` option)

- `-s` Show different sets side-by-side (split screen)
- `-A` For associated events, only show event from first set
- `-B` For associated events, only show event from second set

- `-b=1,2,m` For associated events, take all header lines of the type listed from second set, all other headerlines from first set (implies `-A`). If the second set does not have the required headerline, they will not be included in the output (i.e., specified header lines will never be taken from first set if there is an event association)
CHAPTER 28. MERGE EVENTS IN SEISAN AND COMPARE CATALOGS

1...9,F,E,I,H : the corresponding header lines (Note that line type 4 refers
to phase pick lines, even if they have a blank in column 80
o: : Origin time (from type 1 line only)
e: : Epicentral coordinates (from type 1 only)
d: : Depth (from type 1 line only)
m : copy all magnitude information into header line
mW,mL,mb,mS : only copy designated magnitudes (note that magnitudes are copied
into the same slot unless option -M is set

-a=1,2,... Like -b, but copy information from first set to 2nd set, implies -B (not
implemented yet)

-m=1,2,... Like -b, but add header lines to existing header lines, effectively merging
the information in both files. Other header lines are taken from A (implies -A).

-M Find named slot for copied magnitudes (only relevant if -b={mW,ML,mb,mS} is set
Both type and agency need to match; otherwise the first empty slot is occupied, or
the last slot is overwritten)

-d=xyz Plot differences in hypocentre (in km) (loc2-loc1)
-d=gmt Plot differences in epicentre as gmt-style multi-segment file (for input into psxy)
-d=gmtd Plot differences in hypocentre as gmt-style multi-segment file. This is similar
to the output of -d=gmt, but instead of just latitude and longitude include latitude,
longitude and depth (for pre-processing with awk before passing to psxy)
-d=bias[:<minerr>][:covscale]
Calculate bias vector between the two sets (i.e. the vector that needs to be
added to loc1 locations to make them coincide with loc2 locations on average).
The calculation weighs pairings according to their horizontal errors and an error
estimate is provided for the resulting bias vector. For combined error estimates
less than 1 km (or <minerr>) if set, the weighing is applied assuming the error
is 1 km (minerr). The calculation assumes the area of interest is small such
that all calculations ignore spherical geometry. The average latitude of the
events is used to determine the latitude for the spherical to Cartesian
conversion.

Example Output (actual output without line number)

1 Bias (set B - set A) #eq: 58
2 (Lon,Lat,Dep) = ( 0.01, 0.04, -8.21 )
3 (R,THETA,Z) = ( 4.427799, 19.4, -8.209310 )
4 (X,Y,Z) (km) = ( 1.470831, 4.176369, -8.209310 )
5 (EX,EY,EZ) = ( 0.147675, 0.159429, 0.339260 )
6 CXX,CXY,CXZ,CYY,CYZ,CZZ = 0.0218079551088708, -0.000638279192795213,
   -0.000749725773138676, 0.0254174912655165, 0.00139172907926588,
   0.115097515525919
7 96.1097639008737 2.39858930837142 1.47083143290329 4.17636917834104
   0.0897062757410673 0.0968460018196738 -0.0271104591760068
2 Average shift of events in B with associated events in A in deg latitude, deg longitude and depth (km)
3 As line 2, but horizontal shift is expressed in terms of a distance R (km) and a direction theta (deg)
4 As line 2, but horizontal distances expressed in km rather than degree
5 Formal errors of the mean shift in km. Note that this is the error of the mean and not the standard deviation.
The formal location errors and covariance matrices of the catalogue events are taken into account in this calculation
6 Input line for gmt command psvelo. psvelo expects errors to be specified for 68% confidence intervals.
If covariances are given for a different confidence interval, then they are divided by covscale prior to working out the quantities for psvelo (the error and covariance output lines are always left unchanged). You can use the following table:

<table>
<thead>
<tr>
<th>Confidence</th>
<th>covscale</th>
</tr>
</thead>
<tbody>
<tr>
<td>68%</td>
<td>1.0</td>
</tr>
<tr>
<td>90%</td>
<td>2.71</td>
</tr>
<tr>
<td>95.4%</td>
<td>4.0</td>
</tr>
<tr>
<td>99%</td>
<td>6.63</td>
</tr>
<tr>
<td>99.73%</td>
<td>9.00</td>
</tr>
</tbody>
</table>

[Defaults: minerr=1.0 ; covscale=2.71 (Native Confidence Interval 90%)]

Note: all -d options require -i option

Author: F. Tilmann (tilmann at gfz-potsdam.de) (c) 2002-2011

Examples

1. merge_seisan.pl -s -S=U -T=4 -D=50 cat1.nor cat2.nor
   Display events in both catalogues. The left side (cols 1-80) shows events from the first catalogue, and the right side (cols 83-163) shows events from the second one. Where an event in the first catalogue is within 4 s and 50 km epicentral distance of an event in the second catalogue, the events are displayed side-by-side where corresponding header lines are matched up.
   merge_seisan.pl -s -S=U -T=4 -D=50 DBASE cat2.nor does the same as above but the first catalogue is compiled from all the events in the current database. Instead of DBASE, it is also possible to give an explicit database code.

2. merge_seisan.pl -M -T=10 -i -b=F,mW local.nor cmt.nor > local-with-cmt-mech.nor If cmt.nor is a nordic file with the CMT focal mechanism represented by a 'F' type header line, and magnitude mW set, then the focal mechanism and magnitude information from cmt.nor is combined with all other parameters from local.nor. The -M option forces the mW magnitude to be entered into an empty magnitude slot if available (association requires a time-difference of less than 10 s).

3. merge_seisan.pl -T=10 -i -d=gmt global.nor local.nor > lines.xy produces a multi-segment file suitable as input to gmt command psxy with -m option which will then draw the line connecting the epicentres of associated events in the two catalogues.
CHAPTER 28. MERGE EVENTS IN SEISAN AND COMPARE CATALOGS

4. `merge_seisan.pl -T=10 -i -d=xyz global.nor local.nor > bias.xyz` produces a file suitable for plotting a histogram of the difference between locations in both catalogues.

5. `merge_seisan.pl -T=10 -i -d=xyz global.nor local.nor > bias.xyz` works out the mean shift between events that are present in both catalogues (see usage information above for explanation of output format).

28.5 COMPARE_HYP, compare hypocenters of two cat-files

This program compares hypocenters. The hypocenters can be either from one file (option 1) with several hypocenters for each event and the hypocenters to compare are selected by agency code. The intention with this option is to compare hypocenters from different agencies stored in same s-file. An example is a download from ISC.

Or (option 2) the program compares two Nordic event files with the same events in both. The intention with this second option is to be able to see the effect of locating the same data set using a different model and or different magnitude scales.

Options 2

The input requires two files: The first file is the reference and the second file is the modified reference file. The compared content is origin time, rms, hypocenter and magnitudes. For each parameter, the average difference with standard deviations is calculated. The difference is calculated as the value in the reference minus the value in the modified file. In order to compare the files, they must have the same number of events and it is assumed that the events are the same and come in the same order, however the only thing checked is that the files contain the same number of events. If an event has no data (no hypocenter), the event is skipped. The the list of skipped events is given in a file compare_hyp.skip. The number of magnitudes for each event does not have to be the same so not all events need to have magnitudes. Only magnitudes available in both data sets are compared and listed. The magnitudes are always listed in order Ml, Mc, Mb, MB, Ms, MS, Mw. For the magnitudes there is no check if the agency is the same. If a magnitude appears 2 times, the last one will be used. Note that magnitude in position 3 on header line will not be overwritten when relocating unless it has the agency identical to the agency in the STATION0.HYP file, this can create problems when comparing so make sure the agency in that position corresponds to the agency used when locating. There is no check if epicenter is fixed since normally this is what will be compared. Since the depth very often is fixed, fixed depths are not used for comparison. The output is a file called compare_hyp.out. It lists each event with the data from the reference event and the difference with the relocated event. Events with a difference of more than user defined values for the parameters, have their differences marked with a star, see example below. Events with less than a given number of stations are skipped. Events with no RMS are skipped. Events with flag set for no location are skipped. The skipped events are listed in file compare_hyp.skip. An example run is shown below. The default minimum differences used indicating with a star are 5 km in lat-lon and depth, 1 s in origin time, 0.5 s in rms and 0.2 magnitude units.

Example run

```
c:\seismo\wor>compare\_hyp
Give input file 1, the reference, or single input file
eeve.out
```
Figure 28.1: (Left) Histogram of the offsets between locations in the GCMT catalogue and a catalogue based on an OBS/land deployment offshore Northern Sumatra [Tilmann et al., 2010], generated using the -d=xyz option and GMT pshistogram. (Right) Map and cross-sectional views. Large rimmed circles are events in the local catalogue, and small borderless circles are locations in the GCMT catalogue. Lines connect associated events (drawn using -d=gmtd option and GMT psxy). The red arrow points in the direction of the average shift between CMT and local locations, with the little red circle showing the formal uncertainty of the average shift vector (generated using -d=bias option and GMT psvelo). Its absolute size is only meaningful if the formal errors of the contributing locations are all realistic. The plots show the tendency of CMT centroids in subduction zones to be located seaward off and deeper than their actual location.
Give input file 2, the file to compare
If blank (enter) only one file is used

<table>
<thead>
<tr>
<th>hyp.out</th>
<th>2016</th>
<th>721</th>
<th>1517</th>
<th>46.7</th>
<th>D</th>
<th>41.340</th>
<th>44.040</th>
<th>5.0F</th>
<th>SC3</th>
<th>20</th>
<th>1.5</th>
<th>4.2LSC3</th>
<th>5.7BSC3</th>
<th>4.3bSC3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2016</td>
<td>721</td>
<td>1628</td>
<td>47.7</td>
<td>D</td>
<td>-23.860</td>
<td>-175.560</td>
<td>10.0F</td>
<td>SC3</td>
<td>43</td>
<td>1.9</td>
<td>6.0LSC3</td>
<td>5.8BSC3</td>
<td>5.3bSC3</td>
</tr>
<tr>
<td></td>
<td>2016</td>
<td>721</td>
<td>23</td>
<td>9</td>
<td>3.5</td>
<td>D</td>
<td>40.780</td>
<td>-124.070</td>
<td>10.0F</td>
<td>SC3</td>
<td>33</td>
<td>2.0</td>
<td>4.9LSC3</td>
<td>4.9BSC3</td>
</tr>
<tr>
<td></td>
<td>2016</td>
<td>722</td>
<td>040</td>
<td>26.3</td>
<td>D</td>
<td>39.360</td>
<td>25.970</td>
<td>7.0</td>
<td>SC3</td>
<td>54</td>
<td>1.4</td>
<td>3.1LSC3</td>
<td>3.0LSC3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2016</td>
<td>722</td>
<td>1131</td>
<td>49.5</td>
<td>D</td>
<td>19.900</td>
<td>120.900</td>
<td>10.0F</td>
<td>SC3</td>
<td>17</td>
<td>1.8</td>
<td>5.1BSC3</td>
<td>4.7bSC3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2016</td>
<td>722</td>
<td>2140</td>
<td>10.2</td>
<td>D</td>
<td>42.400</td>
<td>35.210</td>
<td>10.0F</td>
<td>SC3</td>
<td>49</td>
<td>1.0</td>
<td>4.4LSC3</td>
<td>3.9SSC3</td>
<td>4.6BSC3</td>
</tr>
<tr>
<td></td>
<td>2016</td>
<td>722</td>
<td>2344</td>
<td>57.8</td>
<td>D</td>
<td>36.940</td>
<td>26.680</td>
<td>8.0</td>
<td>SC3</td>
<td>15</td>
<td>1.0</td>
<td>2.7LSC3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2016</td>
<td>723</td>
<td>00</td>
<td>20.0</td>
<td>D</td>
<td>47.680</td>
<td>147.050</td>
<td>395.0</td>
<td>SC3</td>
<td>214</td>
<td>1.0</td>
<td>6.9LSC3</td>
<td>5.7BSC3</td>
<td>5.7bSC3</td>
</tr>
<tr>
<td></td>
<td>2016</td>
<td>723</td>
<td>226</td>
<td>20.3</td>
<td>D</td>
<td>-7.410</td>
<td>128.920</td>
<td>132.0</td>
<td>SC3</td>
<td>19</td>
<td>1.1</td>
<td>5.4LSC3</td>
<td>5.3bSC3</td>
<td>5.0bSC3</td>
</tr>
<tr>
<td></td>
<td>2016</td>
<td>723</td>
<td>252</td>
<td>12.1</td>
<td>D</td>
<td>30.140</td>
<td>142.110</td>
<td>61.0</td>
<td>SC3</td>
<td>14</td>
<td>0.7</td>
<td>4.9LSC3</td>
<td>4.9BSC3</td>
<td>4.8bSC3</td>
</tr>
<tr>
<td></td>
<td>2016</td>
<td>723</td>
<td>741</td>
<td>45.1</td>
<td>D</td>
<td>-6.710</td>
<td>101.270</td>
<td>10.0F</td>
<td>SC3</td>
<td>83</td>
<td>1.5</td>
<td>6.0LSC3</td>
<td>5.8BSC3</td>
<td>5.3bSC3</td>
</tr>
</tbody>
</table>

End of s-file

Min difference to mark, lat-lon, depth, origin, rms, mag
Defaults (enter) are 5km, 5km, 1s, 0.5s, 0.2s

Min number of stations to compare, def 0

<table>
<thead>
<tr>
<th>Origin time RMS</th>
<th>Lat</th>
<th>Lon</th>
<th>Depth</th>
<th>Ml</th>
<th>Mb</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average diff</td>
<td>0.4</td>
<td>0.0</td>
<td>-0.005</td>
<td>-0.001</td>
<td>0.5</td>
<td>-0.1</td>
</tr>
<tr>
<td>Standard dev</td>
<td>1.0</td>
<td>0.1</td>
<td>0.033</td>
<td>0.010</td>
<td>4.5</td>
<td>0.1</td>
</tr>
<tr>
<td>Number of values</td>
<td>10</td>
<td></td>
<td>11</td>
<td>6</td>
<td>10</td>
<td>8</td>
</tr>
</tbody>
</table>

Output file name is compare_hyp.out
Output of events deviating is compare_hyp_dev.out
Output of events ok is compare_hyp_ok.out

The last two files can be used to plot bad and good events respectively.

In textttcompare_hyp.out, all the input headers are listed for the two files and at the end a summary of the differences is given. In the example above, note that there were 11 events. Five events had fixed depth so only 6 were used for depth comparison. For RMS, only 10 out of the 11 has an RMS. Also note that not all events had all the magnitude types. This example is comparing locations done with SeisComp3 and SEISAN.
NOTE: If RMS is 0.0, it is assumed there is no rms and it is not used. Below is shown the output file.

<table>
<thead>
<tr>
<th>#</th>
<th>Origin time RMS</th>
<th>Lat</th>
<th>Lon</th>
<th>Depth</th>
<th>N</th>
<th>Ml</th>
<th>Mb</th>
<th>MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20160721151746.7</td>
<td>1.5</td>
<td>41.340</td>
<td>44.040</td>
<td>5.0</td>
<td>20</td>
<td>4.2</td>
<td>4.3</td>
</tr>
<tr>
<td></td>
<td>0.2-0.1</td>
<td>-0.002</td>
<td>-0.013</td>
<td>0.0</td>
<td>-0.3*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20160721162847.7</td>
<td>1.9</td>
<td>-23.860</td>
<td>-175.560</td>
<td>10.0</td>
<td>43</td>
<td>6.0</td>
<td>5.3</td>
</tr>
<tr>
<td></td>
<td>0.1 0.1</td>
<td>0.013</td>
<td>-0.012</td>
<td>-0.1</td>
<td>0.1</td>
<td>-0.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20160721230903.5</td>
<td>2.0</td>
<td>40.780</td>
<td>-124.070</td>
<td>10.0</td>
<td>33</td>
<td>4.9</td>
<td>4.7</td>
</tr>
<tr>
<td></td>
<td>0.3 0.0</td>
<td>0.010</td>
<td>0.000</td>
<td>0.0</td>
<td>-0.3*</td>
<td>-0.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>20160722004026.3</td>
<td>1.4</td>
<td>39.360</td>
<td>25.970</td>
<td>7.0</td>
<td>34</td>
<td>3.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.8 0.0</td>
<td>-0.005</td>
<td>-0.018</td>
<td>-0.3</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>20160722113149.5</td>
<td>1.8</td>
<td>19.900</td>
<td>120.900</td>
<td>10.0</td>
<td>17</td>
<td>4.7</td>
<td>5.1</td>
</tr>
<tr>
<td></td>
<td>0.1 0.1</td>
<td>-0.011</td>
<td>0.005</td>
<td>0.0</td>
<td>-0.2*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>20160722214010.2</td>
<td>1.0</td>
<td>42.400</td>
<td>35.210</td>
<td>23.0</td>
<td>49</td>
<td>4.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-1.7*0.3</td>
<td>0.057*</td>
<td>0.004</td>
<td>0.8</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>20160722234457.8</td>
<td>1.0</td>
<td>36.940</td>
<td>26.680</td>
<td>8.0</td>
<td>15</td>
<td>2.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-0.7 0.1</td>
<td>-0.011</td>
<td>-0.003</td>
<td>-2.5</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>20160723010020.0</td>
<td>1.0</td>
<td>47.680</td>
<td>147.050</td>
<td>395.0</td>
<td>214</td>
<td>6.9</td>
<td>5.7</td>
</tr>
<tr>
<td></td>
<td>0.7 0.2</td>
<td>-0.092*</td>
<td>0.010</td>
<td>9.2*</td>
<td>-0.4*</td>
<td>0.3*</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>20160723022620.3</td>
<td>1.1</td>
<td>-7.410</td>
<td>128.920</td>
<td>132.0</td>
<td>19</td>
<td>5.4</td>
<td>5.0</td>
</tr>
<tr>
<td></td>
<td>0.0 0.0</td>
<td>-0.001</td>
<td>-0.004</td>
<td>-3.3</td>
<td>-0.2</td>
<td>0.2</td>
<td>0.3*</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>20160723025212.1</td>
<td>0.7</td>
<td>30.140</td>
<td>142.110</td>
<td>61.0</td>
<td>14</td>
<td>4.9</td>
<td>4.8</td>
</tr>
<tr>
<td></td>
<td>0.0 0.0</td>
<td>0.000</td>
<td>0.013</td>
<td>-0.9</td>
<td>-0.2</td>
<td>0.4*</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>20160723074145.1</td>
<td>1.5</td>
<td>-6.710</td>
<td>101.270</td>
<td>10.0</td>
<td>83</td>
<td>6.0</td>
<td>5.3</td>
</tr>
<tr>
<td></td>
<td>0.0 0.0</td>
<td>-0.009</td>
<td>0.008</td>
<td>-0.1</td>
<td>0.1</td>
<td>-0.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For each event, the difference in value is shown. Note that the difference value has a star behind it if it exceeds the user specified values. The event numbers are also listed.

Option 1, a single file

Comparison of hypocenters is done between the hypocenters listed for the same event in the s-file. The hypocenters are selected based on user specified agencies. There is no comparison of magnitudes. An output file of the events having hypocenters from specified agencies is made. In addition 2 compact file are made of the hypocenters and magnitudes from each of the two agencies making it possible to e.g. plot the hypocenters separately for the two agencies. Only magnitudes from corresponding agencies are selected, they are limited to 3. The output of the comparison in the standard output file as for option 2. is the same as for option 2. Below is an example run.

```
compare_hyp
Give input file 1, the reference, or single input file
```
test.out
Give input file 2, the file to compare
If blank (enter) only one file is used

Give reference agency
ISC
Give agency to compare to
TIR

1982 3 9 2311 34.9 R 39.825 19.879 12.7 ISC 67 1.8 4.5bISC 4.3SISC 4.5sISC1
1983 3 3 931 16.2 R 40.155 19.797 9.00F ISC 97 2.1 4.5bISC 4.4SISC 4.5bISC1
1995 524 340 27.8 RQ 39.711 20.437 7.00 ISC 36 1.7 3.6bISC 3.5bNEI 3.6bISC1
1999 112 726 11.0 RQ 40.024 20.528 10.0 ISC108 1.7 4.1bISC 4.0bNEI 4.1bISC1
1999 3 9 341 5.1 R 39.195 20.812 9.50 ISC 59 1.8 3.6bISC 3.8bNEI 3.6bISC1
1999 3 9 523 3.1 R 39.171 20.846 31.8 ISC 65 1.7 3.7bISC 4.0bNEI 3.7sISC1
1999 412 1716 28.0 R 40.413 21.450 13.2 ISC 86 1.8 3.7bISC 3.9LPDG 3.7bISC1
1999 424 723 18.9 RQ 41.448 19.536 20.4 ISC118 2.4 4.2bISC 4.0bNEI 4.2bISC1

Origin time RMS Lat Lon Depth
Average diff -0.6 1.1 -0.089 0.053 -0.5
Standard dev 4.8 0.5 0.299 0.272 11.0
Number of values 75 106 99

Output file name is compare_hyp.out
File with events with both agencies is compare_hyp_both.out
File with only events from ISC is ISC.out
File with only events from TIR is TIR.out
Chapter 29
Making synthetic seismograms

BOUCH and BOUSEI, HERRMANN and HERRSEI and WKBJ are all programs which is used for generating synthetic seismograms.

The full wave modeling programs are written by Bouchon and Herrmann, and for WKBJ, Chapman and Valerie Maupin. Valerie Maupin has integrated WKBJ for SEISAN and written the routines that makes it possible to use specific phases. She has also made many improvements in the original installation of BOUCH and HERRMANN and written a large part of this chapter.

Bouchon:
The Bouchon program is somewhat modified for SEISAN. The theory, which is quite straight forward, is given in a series of papers (e.g. Bouchon [1981]). It is based on a discrete wave number representation of the wave fields. Basically, the source is repeated periodically in space, so that integration over the k-domain is replaced by a series. This implies that the periodicity of the source, L (in km), should be large enough so that the information from fictitious sources does not arrive during the time interval of interest. Roughly r < L/2, sqrt((L-r)^2+Z^2) > Vp*t where r is the epicentral distance and Vp is the highest P-wave velocity of the model, t is the travel time and Z the hypocentral depth. Only layered (horizontal, parallel) earth model is used. The earthquake source cannot be in the bottom layer or at the surface.

There are 2 programs, BOUCH and BOUSEI. BOUCH computes the frequency response given the model, the source depth, the focal mechanism, the receiver locations and the orientations of the two horizontal components. BOUSEI takes the output file from BOUCH, multiplies it by the source spectrum and uses an FFT to get the synthetic ground motion (displacement, velocity or acceleration). The user must provide the source function (see below) and the original waveform files must be available in WAV or working directory if a file containing both real and synthetic signal is to be generated. Otherwise, only synthetic data will be seen in the output file.

Herrmann:
The Herrmann programs HERRMANN and HERSEI work the same way as BOUCHON and BOUSEI respectively. The major difference is that once HERRMANN has been executed, HERSEI can be executed with different fault plane solutions to obtain the time series, while for the Bouchon programs, both programs must be run again. The Herrmann programs are thus faster for testing many different fault plane solutions.

The description in the following is for the Bouchon programs, but the steps are the same for HERRMANN.
WKBJ:
As opposed to the seismograms calculated with the Bouchon and Herrmann programs, the WKBJ synthetic seismograms contain only the number of phases selected by the user. The execution time for one run of the program is very short. In addition to making the synthetic seismograms, the program calculates the arrival times of these phases, and write them both on the screen and in the iasp.out file for later plotting (see MULPLT). This is intended to be a tool to help identify phases on the data or on the Herrmann or Bouchon synthetic seismograms: it can by no means replace these two programs, which are much better than WKBJ to model the frequency-dependent character of crustal phases at regional distance.

WKBJ seismograms have been introduced in seismology by [Chapman 1978]. More details on the method can be found in [Dey-Sarkar and Chapman 1978] and in [Chapman and Orcutt 1985]. The core of the present program is a code written by [Chapman et al. 1988] and is part of the seismological software distributed freely by IASPEI. The synthetic seismograms are given in displacement. Although their spectra contain low frequencies, one should bear in mind that they represent a high-frequency approximation of the wave field. They include a number of non-physical phases due to truncation of the integrals in slowness \( p \). For the most interesting crustal phases, the epicentral distance is usually much larger than the source depth, and these phases interfere with the physical phases and modify their amplitudes.

The head waves on an interface appear automatically as a by-product of the reflected phases, as soon as the epicentral distance is larger than critical. That means for example that the Pn phase appears automatically on the synthetic seismogram as a by-product of the PmP phase. In order to synthesize or calculate the arrival time of a Pn or Sn phase, you must then specify 'PmP' or 'SmS' (see below).

For a receiver at the free surface, the synthetic seismograms must include the free surface reflection coefficient to yield correct amplitude and waveform for the different phases. For S phases, at epicentral distances larger than critical, this includes automatically the SP phase (a P phase which propagates horizontally along the free surface, and which originates from the critical conversion of S to P at the free surface). The critical distance is of the order of the source depth for the Sg phase, and its SP phase usually appears as a large arrival between the P and S wave. The SP phases are physical, but the amplitude of their high frequency part is overestimated with WKBJ. If one wishes to suppress them from the synthetic seismograms, one may optionally do so. With this option, the surface reflection coefficient is omitted and the synthetic seismograms contain only the upgoing wavefield, that is the wavefield one would get in a borehole, after filtering out the downgoing wavefield. Let us note that this option may strongly modify the amplitudes and waveforms of the different phases compared with those at the free surface.

In addition to the synthetic seismograms, the program calculates the arrival times of the phases you have specified, and write them in the iasp.out file. These times are calculated by interpolation in epicentral distance of the values tabulated in wkbj.tab. For sources close to an interface (in practice for Pg and Sg phases and the source under an interface), there is a limited epicentral distance range in which an arrival time can be calculated. For example, the maximum epicentral distance for Pg is about 250km for a source 0.1 km under Moho in the default SEISAN model. In order to increase the maximal epicentral distance, you may move the source away from the interface, or you may increase the number of ray parameters used in program wkbj_or.for (parameter 'mapp') called from wkbj.for.

All three programs are hardwired to use triangular sources.

Running the programs
The programs require input about distances, azimuths, depth, crustal model, fault plane solution, time window, number of points and some modeling parameters. Almost all of these parameters are available within SEISAN. The programs have therefore been modified to use an S-file (Nordic format) as input file with additional information about time window, number of points to model and crustal model. A special
format has been used to keep the modeling information separate from other information in the file (see below for an example). The steps to model a particular event are as follows:

Problem Bouchon: Use fewer layers, ideally just a halfspace under the deepest ray. The programs seem to become unstable if too many layer are used there.

Step 1
Edit the event in EEV and mark the stations wanted for modeling with a minuscule s in column 1, ONLY mark the station once. Exit from editor and, within EEV, give the command "synt". This will generate all the necessary default input parameters for modeling, which are stored as comment-lines starting with SYNT in the S-file (see below). At the same time, the s’s used as markers are removed. Any old modeling information present will remain and override the defaults. However, in case the F-flag is set for the DEPTH parameter, distances and azimuths will be reset according to the current location.

Step 2
Edit event again and check if default parameters are ok (see explanation below).

Step 3
Run one of the programs BOUCH, HERRMANN or WKBJ. These are known commands in EEV. BOUCH: The program will now run for a certain amount of time depending on number of points required. At the standard output, the input parameters used will be printed out and for each frequency, the number of terms in wave number integration is printed out. If the limit of the number of terms is reached, something is wrong, try other parameters. The limit is 2. BOUPAR parameter, currently set at default value of 2000. The speed of this output (NPOINT/2+1 lines) gives a good indication of how long time it will take. HERRMANN: Takes longer than BOUCH. WKBJ: Very fast.

Step 4
Generate the seismograms. BOUCH: Use program BOUSEI. The program is interactively asking the seismogram type (displacement, velocity or acceleration). BOUSEI will generate a file bousei.out in SEISAN format containing both original and synthetic traces. The number of traces is determined by the specifications for each station, see below. Output file is bousei.out. HERRMANN: Use program HERSEI, similar to BOUSEI. Output file is hersei.out. WKBJ: The first command WKBJ also makes the seismograms. Output file is wkbjsei.out.

In all cases it is possible to shift the original trace relative to the synthetic trace and the program will ask, for each channel, how much it should be shifted. A positive value shifts the real trace up in time (to the left). The default is to shift the trace the amount of the P-travel time residual of the first P found in the S-file for that station in order to line up the P - phases. NOTE: These phases MUST be the same phase types in order to be lined up. If the first modeled phase is Pn and the first observed phase given in the S-file is Pg, there will be a no alignment. The amplitudes for Bouchon are in nm, nm/sec or nm/sec*sec (hopefully !!) assuming a seismic moment of 10 **22 dyne-cm. The output file will normally contain both the original and synthetic traces. However, if no waveform file is available (in local or WAV directory), the output file will contain an empty channel where the original data should have been. The specifications in the hyp.out file determine which traces from the modeled stations are included in the output file. If the specification after STATION is only component (e.g. S), then all 3 channels are shown. If a particular channel is given (e.g. S N), then only that channel is shown. Only one or 3 channels can be displayed. All output traces are given in Z, N and E or Z, R and T depending on the parameter file (see below). The channel names are SH, SB and SW for Herrmann, Bouchon and WKBJ respectively.

Step 5
Plot the traces with mulp. This can be done within EEV using the command pw, ph or pb for WKBJ, Herrmann or Bouchon respectively. Since there is no instrument correction, it is a good idea to plot both the modeled and observed signals narrow band pass filtered. E.g. for regional events 0.1-1 Hz and for small local events 2-5 Hz (depending on sample rate).

shows an example of the modeling.

Note: The whole modeling process can be done entirely within EEV and it is intended to be done so. Since the modeling requires updated distances, depths etc when changing model etc, it cannot take its input from the location in the S-file, which only changes when doing an update (see UPDATE program). So when running from within EEV, a location will always be done first to get an updated S-file (in this case the hyp.out file) and this is the reason that the modelling programs use the hyp.out file instead of the S-file for input. This also means that the modeling program can be run separately from any hyp.out file, however it is then up to the user to keep it updated.

The modeling parameters

Below is shown an example of part of an S-file prepared for modeling. The file is one of the events in the test data set and by using EEV to find the event, modeling can start immediately. All parameters have been set automatically.

```
1996 6 7 1325 29.1 L* 59.841 5.127 12.0F BER 12 1.1 2.2CBER 1.9LBER 2.0OLNA01
1996 6 7 1325 29.1 L* BER 2.0WBER 1.9LBER 2.0OLNA01
8.3 41.0 74.7 1 F
1996-06-07-1324-51S.TEST__009
535 SOUTHERN NORWAY
depth fixed to 12 km, rms lowest with near station (less than 110 km) location
fault plane solution ok within 10 deg
1996 6 7 1325 29.1 L* 59.841 5.127 12.0F BER 12 1.1 2.2CBER 1.9LBER 2.0OLNA01
1996 6 7 1325 29.1 L* BER 2.0WBER 1.9LBER 2.0OLNA01
8.3 41.0 74.7 1 F
1996-06-07-1324-51S.TEST__009
535 SOUTHERN NORWAY
depth fixed to 12 km, rms lowest with near station (less than 110 km) location
fault plane solution ok within 10 deg
```

SYNT: MODEL---: THICK VP VS DENS QP QS
SYNT: MODEL---: 12.000 6.200 3.563 2.600 0.000 0.000
SYNT: MODEL---: 11.000 6.600 3.793 2.800 0.000 0.000
SYNT: MODEL---: 8.000 7.100 4.080 3.000 0.000 0.000
SYNT: MODEL---: 19.000 8.050 4.626 3.200 0.000 0.000
SYNT: MODEL---: 30.000 8.250 4.741 3.400 0.000 0.000
SYNT: MODEL---: 50.000 8.500 4.885 3.600 0.000 0.000
SYNT: ST-D-RK: 8.3 41.0 74.7
SYNT: DEPTH---: 12.0
SYNT: NPOINTS: 512
SYNT: TIMES---: TOTAL 60.000 INITIAL 0.000 SY-TRACE 60.000
SYNT: BOUPAR--: 800.0 2000 0.010
SYNT: PHASES--: Pg Sg PmP SmS SmP
SYNT: DT-Tsou: 0.050 .100
SYNT: REDVELO: 0.0000
SYNT: COMPON--: RADIAL
SYNT: STAT-AT: no
SYNT: NSTAT---: 3
SYNT: STATION: EGD S Z DISTANC: 49.0
SYNT: STATION: EGD AZIMUTH: 9.0 BAZIMUTH: 189.2
SYNT: STATION: KMY S Z DISTANC: 71.0
SYNT: STATION: KMY AZIMUTH: 172.0 BAZIMUTH: 352.7
MODEL: The model to be used. THICK is layer thickness, VP is Vp velocity, VS is Vs velocity, DENS is density and QP and QS, are P and S q-values respectively. The model, velocities and Q-values are taken from the STATION0.HYP file with first choice from current directory and second choice from DAT directory (like the HYP program). The S-velocities are calculated using the Vp/Vs ratio given there. Moho is indicated with N at the end of the line with the first mantle layer. A Q of zero means infinite Q. The densities are approximate values and should be modified. See below for maximum number of layers.

ST-D-RK: Strike dip and rake is taken from an existing fault plane solution for the given event (F-line) if it exists, otherwise arbitrary values are supplied. (0,0,0) is an explosion. The convention is Aki and Richards.

DEPTH: Focal depth is taken from the current solution. The second field can optionally have the letter F (right justified). If this flag is set, the user can give the synt command to update all distances and azimuths used for modeling which will correspond to the latest location determined as e.g. a result of a changed fixed depth or a changed model. The intention with this flag is that the user should be able to set a fixed depth in the S-file header line, give the synt command to update the parameters for modeling corresponding to this depth and then model.

NPOINTS: Number of points to model, 512 is set as default, must be 2**N. Used by BOUCH and HERRMAN only.

TIMES–: Three different times:
TOTAL: The total time window for generating data and synthetic seismograms for all channels, see also REDVELO.
INITIAL: The initial time of the earliest trace in the output file, with reference to the source origin time. The synthetics at the station with smallest epicentral distance automatically start also at this initial time.
SY-TRACE: The duration of the synthetic seismogram for each channel, might have different start times, see REDVELO.

DT-Tsou: Sampling interval (used for WKBJ seismograms only), and half-duration of the source used for all three programs. In all programs, the source is triangular, however BOUCH can optionally use several sources, see below.

REDVELO: Reduction velocity to calculate the initial times at subsequent distances (put 0. for no reduction velocity). NOTE: Seems to not be correctly implemented so use 0 always.
CHAPTER 29. MAKING SYNTHETIC SEISMOGRAPHS

PHASES-: The names in format A4 (right justified) of the phases to be synthesized with WKBJ. The phases may be given in any order, with a maximum of 6 phases per line, and there may be several “SYNT: PHASES-” lines.

Possible phases:

- Pg (direct P from source to receiver)
- Sg (direct S)
- PmP (includes automatically Pn at distances larger than critical)
- pPmP (includes automatically pPn at distances larger than critical)
- sPmP (includes automatically sPn at distances larger than critical)
- SmS, pSmS, sSmS (includes automatically Sn, pSn, sSn at distances larger than critical)
- SmP, PmS
- P1P, P2P, S1S, etc: the same as PmP, SmS etc, but on interface number 1, 2, etc.

(The free surface gets interface number 0 in the convention taken here. Thus in HYP, PN2 is the same as P1N here. There associated head waves are labeled Pn1, Pn2, Sn1, etc.

COMPON-: RADIAL for radial-transverse components, NORTH for North-South, East-West components.

STAT-AT: Is “not free” or “NOT FREE” anywhere within column 16 to 25: Optional line. If this option is chosen, the WKBJ synthetic seismograms are calculated omitting the reflection coefficient at the free surface, at the receiver location.

BOUPAR: Modeling parameters L, Nt and e. L is length of periodicity (should be a few times the hypocentral distance), Nt is maximum number of terms in wave number summation and e is the value used in truncating the summation. Increasing e and decreasing Nt will speed up convergence, but the results might be unreliable. If Nt is reached, the results are unreliable.

NEW STAT: Comment line

STATION: Station to be modeled with component(s) to be displayed. The S means that short period instruments are used. The default is S, so if e.g. BH is used, S must be changed to BH, else the waveform data is not found. If no component is given, all 3 components are assumed. The other option is to indicate a component (e.g. Z) and only that component will be displayed (see also description of BOUSEI). DISTANC is epicentral distance used, this distance is taken from the current location, AZIMUTH is azimuth from the source to the station taken from current location, BAZIMUTH is the back azimuth at the station, calculated by EEV, used to rotate if so specified. Each new station is represented by the above 3 lines.

The source time function

The time duration of the triangular source time function for Bouchon is given as Tsou above, and is also used in WKBJ and Herrmann.

Hints on modeling

Event 199606071325 in the test data set is set up with modeling parameters and can be tested immediately.

The model

The standard model given in STATION0.HYP might be too detailed for most cases and should be simplified to include 3-4 layers by just editing the S-file, this also speeds up modeling. However, if you located the event with one model and model with another, the distances and residuals might not fit. A solution could be to have a STATION0.HYP in the local directory with the simplified model.

Alignment of P and S
If the distance calculated by HYP is not correct as indicated by P and S residuals, the synthetic and observed signals will not be aligned. The distance for that station can then be changed manually in the S-file under DIST and/or delays can be applied when generating the seismograms. For line up, it is important that the correct first arrival is included in phase list (WKBJ), see what is identified by HYP. If PN2, then P1N must be given for WKBJ.

Testing different parameters

There is no need to go back to EEV to test for the parameters that do not change the location. Thus to test for different fault plane solutions, time windows, number of points, edit the hyp.out directly and rerun. However, if depth or model is changed, relocation must be made. To test for different depths, locate with fixed depths, see HYP.

NOTE: THE SOURCE AND RECEIVER CANNOT BE AT THE SAME DEPTH (BOUCH AND HERRMANN) AND IN NO CASES CAN THE SOURCE BE AT DEPTH ZERO.

Running time

This depends mostly on the number of points and to some degree on number of layers. The number of stations has an insignificant effect on running time.

Program limitations: HERRMANN and WKBJ is set up with max 20 layers and Bouchon with 20 layers. Maximum of 32 stations Change programs and recompile if more layers are needed. Bouchon is compiled for 2048 frequencies (4096) points.

Computer notes:

The original Bouchon program BOUCH is almost unchanged. The only modification is that it uses a subroutine to generate its original input file bouch.inp from the hyp.out file. This file still remains after running BOUCH for debugging purposes. The output from BOUCH is bouch.out, which in turn is input to BOUSEI.

Herrmann:

The Herrmann waveform modeling is based on a concept where the synthetic seismograms are computed through a sequence of four distinct processes (programs).

1. The program "hspec8" will calculate the medium response for 10 basic Green’s functions, where the response is given in frequency - wavenumber domain F(f,k).
2. The program "rhwvinta" will integrate and take the medium response from F(f,k) \rightarrow F(f,r)
3. The program "rhfoc10" will convolve the response function with a source time function and with inverse Fourier transform take F(f,r) \rightarrow F(t,r)
4. The program "mech" will construct a 3 component synthetic seismogram given a focal mechanism.

Herrmann’s programs originally had several optional source time functions, however, a triangular source has been hardwired (for all 3 programs) so it is easier to compare the results. The original options can be reactivated by editing the program.

The programs HERRMANN and HERSEI run these 4 programs in an automated sequence.

All References, a detailed manual, source code and parameters as well as other related programs: “Computer Programs in Seismology”, Volumes I - VIII. By Robert B. Herrmann, Saint Louis University, Saint Louis, Missouri.

WKBJ:
CHAPTER 29. MAKING SYNTHETIC SEISMOGRAMS

Input file hyp.out, read by "WKBJ.
Output file iasp.out, written by "WKBJ". Contains the arrival times of
the different phases at the stations, in SEISAN format.
Output file wkbjsei.out, written by WKBJ in the SYNTSEL.FOR subroutines. A waveform file
(SEISAN type) containing the data and the synthetics, which can be plotted using "mulplt". Note
that there is a code for each synthetic seismogram giving the modeling method (SH: Herrmann, SB:
Bouchon, SW: WKBJ), and the component (Z, R, T, N or E).

INTERMEDIATE FILES
wkbj.inp, created by WKBJ for input to WKBJ.OR. The same information as in hyp.out, in a
WKBJ.OR format.
wkbj.tab, output from WKBJ.OR, reprocessed by WKBJ. Contains tables as a function of ray param-
eter.
wkbj.out, output from WKBJ.OR, reprocessed by WKBJ. Contains the Green functions.
Figure 29.1: An example of synthetic seismograms using Bouchon (2), Herrmann (3) and WKBJ (4). The original seismogram is shown in channel 1. All synthetics are displacement. Also shown are the theoretical travel times calculated by WKBJ.
Chapter 30

Calculation and plotting of travel times

In SEISAN, travel times are generated from a flat crustal model or using the IASP91 global travel time model. It can often be useful to generate travel times for given distances and two programs are supplied to do these calculations. TTIM will calculate travel times for global phases at one given distance and depth and TTLAYER, will calculate a travel time table (layered flat model) for a given depth and a distance range. A special version of TTIM called IASP is used in connection with EEV and MULPLT.

30.1 IASPEI travel time software, program TTIM

This program can be used for calculating global travel times, see below for details on phases calculated. The program assumes that you have the travel time tables in the working directory or in DAT, see computer notes below on how to generate these file if not already there. The same files are also used by HYPOCENTER.

After starting the program, the first two questions 'do you want xxxxx' relate to range summaries, etc., that are normally not required and can be answered with n(no) followed by ENTER. The program then asks 'Enter phases, one per line...' You can then enter a specific phase, or a keyword defined as follows:

- **All** gives all phases
- **P** gives P-up, P, Pdiff, PKP, and Pkikp
- **P+** gives P-up, P, Pdiff, PKP, Pkikp, PcP, Pp, Ppdiff, PPKP, PP-KIKP, Sp, Spdiff, SPKP, and SPKIKP
- **S+** gives S-up, S, Sdiff, SKS, Ss, Ssdiff, SSKS, Ps, Psdiff, and PSKS
- **basic** gives P+ and S+ as well as ScP, SKP, PKKP, SKKP, PP, and P’P’

Writing all individual phases, separate by ENTER, terminating the list with an additional ENTER. The program will then enter a loop where phase times are calculated for new distances entered on request. The program is terminated for a particular distance by entering -1, and a new depth can be used, or the program can be terminated by entering -1 again.

A special version of this program used in connection with MULPLT and EEV is IASP. When used from MULPLT, it will use an input file with stations to use called isasp.inp. The file contains station codes of
CHAPTER 30. CALCULATION AND PLOTTING OF TRAVEL TIMES

all station in waveform file(s). If no iasp.inp present, it will use all stations in S-file. indexiasp.inp

In order to generate the earth model files IASP91.HED and IASP91.TBL, first run program REMODL, then program SETBRN. The program REMODL has the earth model hardwired. Note: These binary files CANNOT be moved between platforms. They are included with SEISAN for each respective distribution. If lost, they must be regenerated on the same platform.

For more information about IASP91 programs, see HYPOCENTER manual by B. Lienert.

On at 64 bit computer the IASP files must be regenerated is you have the files from a 32 bit computer, with the programs REMODL and SETBRN.

30.2 Calculation of travel times for layer and gradient model, TTLAYER

The TTLAYER program is written by Barry Lienert to calculate travel times for both layer and gradient model. In this version the program only works for zero depth, and therefore might not be very useful. The program reads a set of velocities and depths from an input file in ‘STATION0.HYP’ format and calculates travel times for P and S velocities for a set of uniform-velocity layers, using the HYPOCENTER dtdx2 routine and also for a set of uniform gradient layers, using dtdxg, a new routine written to have the same input arguments as dtdx2.

The routine to calculate travel times for a gradient model uses an adapted version of Fred Klein’s TTCAL routine, which he uses in his program TTGEN to generate a table of values from which to interpolate travel times and their derivatives in HYPOINVERSE.

The program is easy to run and the output can be plotted with some standard xy plotting tool.

30.3 Calculating of local phases: TTLOCAL

The TTLOCAL program is started from MULPLT to compute local travel times for all stations with waveforms. The program uses the HYPOCENTER travel time calculations. By default, the model is used from STATION0.HYP, unless another model is selected in the s-file.

30.4 Plotting of travel times, TTPLOT

Program to plot observed and calculated travel times (Figure 30.1). The input to the program is an s-file, which has an indicator to a model file (STATION?.HYP) and the travel time observations. The program is started by ‘ttplot <sfile-name>’. At the start, TTPLOT relocates the event and calculates distances using the HYPOCENTER program. It then plots all observations with a ‘+’ symbol and the theoretical travel times that are calculated by the program for the first P and S arrivals with solid lines.

The program can be useful in routine processing to visualize large residuals, which otherwise are seen from the location program output. The program can also be started from EEV using option ‘ttplot’. It is possible to click on symbols, which will bring up station code, phase, observed travel time and residual on the right. The output files are:

- ttplot.out - gives station code, phase name, distance, observed travel times and residual.
- ttplot.eps - Postscript version of plot.
Figure 30.1: Example of travel time plot. Both P and S observed travel times are plotted with "+" symbol. Calculated times are shown by the solid lines, the lower one gives the first P arrivals, the upper line gives first S arrivals. The two outliers are observations from a station with incorrect timing.
30.5 IASP, travel times for MULPLT

This program is a special version of IASP91 to be used in connection with EEV and MULPLT. Giving command iasp from the EEV prompt (or from within MULPLT), the program will read the current active S-file, and for each station, calculate possible IASP91 phases and arrival times relative to the hypocenter and origin time given in S-file. The origin information can be obtained from two places in the S-file: (1) The header lines are searched for hypocenter lines and the first found after the main header will be used, (2) If no secondary header lines, the main header line is used. The intention of this order is that it is possible to put in a PDE solution in a secondary header line (option INPUTONE in EEV) so that theoretical travel times are calculated relative to a fixed solution and not the temporary solution made by the local agency.

The IASP91 tables can be found in the local directory or DAT and have the same names as used in HYP and TTIM. The program generates an output file iasp.out in Nordic format. This file is read by MULPLT and the theoretical phases displayed on the screen. The number of phases calculated can be very large making it hard to see which phase is which. IASP therefore has a definition file, IASP.DEF, where phases to be written out are given. The file can be in the working directory or in DAT. If no definition file is available, all phases will be written to the iasp.out file. Below is an example of a IASP.DEF file.

This file contain the definitions of phases to be used when calculating synthetic phases to be plotted with mulplt. There is one phase pr line and each phase is preceded with the keyword IASP-PHASE. Only lines with this keyword will be read. The defined phase then follows in column 13 to 20. If no phases are defined, all ISPEI91 phases will be used.

<table>
<thead>
<tr>
<th>Phase ID</th>
<th>Phase---</th>
</tr>
</thead>
<tbody>
<tr>
<td>IASP-PHASE P</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE PP</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE PPP</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE PKP</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE pP</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE sP</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE PcP</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE S</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE SS</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE SSS</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE SKS</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE ScS</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE PS</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE SP</td>
<td></td>
</tr>
<tr>
<td>IASP-PHASE ScP</td>
<td></td>
</tr>
</tbody>
</table>

Notice that the definition file might prevent display of phases expected. In the example above, e.g. Pg would not be seen so it is important to set up the definition file for the phases wanted. For local earthquakes, it is not enough to write P or S since these phase are named Pn, Pg, Pb, Sn, Sg and Sb so these phases names must be written like that.
Chapter 31

Inversion for $Q_{Lg}$, QLG

The QLG program can be used to determine an average $Q_{Lg}$ or to perform a tomographic inversion. The method is described in Ottemöller et al. [2002]. Here, we use the same names for the damping parameters, and many of the other parameters should be self-explanatory. The program can also produce the input for distance trace plots. Note that using the program is no trivial task. The data set needs to be carefully selected and the instrument calibration has to be known. The input to the program is a Nordic file, which includes several events. The parameter file needs to be carefully set up.

The program can be used in the following way:

1. Determine average $Q_{Lg}$
2. Perform checker-board test to chose damping parameters
3. Tomographic inversion

Note: The main purpose of including the program is to give an example source code so that the user can make use of it when implementing similar programs. The program uses a linear grid...

Example of the parameter file qlg.par:

```
KEYWORD............Comments.............Par 1......Par 2......
FILTER................for distance plot 0.01 15.
DISTANCES............min and max 200. 3000.
GROUP VEL LG.......lg group vel window 3.0 3.7
GROUP VEL P........p group vel window 5.0 8.0
INVERSION TYPE.....1. for tomography 1.
                    0. for average
ORIENTATION.........0.=vert, 1.=horiz. 0.
PHASE ONLY..........1.=phase pick requ. in s-file 1.
FREQUENCY...........frequency and 1/q 1. 5.2392E-03
```

437
Menke et al. [2006] pointed out the non-uniqueness in attenuation tomography between the source term and Q. They suggest to investigate the non-uniqueness by synthetic tests in which a perturbation is applied to the source term and the inversion for Q is done without inverting for differences in the source term. The solutions obtained are null-solutions and one needs to be careful not to mistake them for real patterns. These tests are possible within QLG by setting the parameter ‘SOURCE PERTURBATION, where the first parameter refers to the source that is perturbed and the second parameter gives the amount of perturbation in units of moment magnitude.

It is possible to invert real data without inverting for the site term by setting ‘FIX SITE’. This can be a useful test as there is also a trade-off with the site term. Fixing the site term is more problematic, as this is done based on the local magnitude, which may not be the same as the moment magnitude.

Another useful stability test is to add Gaussian noise to the spectra and check the inversion result. This can be done for both real data and the checkerboard test by setting the parameter ‘GAUSSIAN NOISE’, units are equivalent to change in moment magnitude.
Chapter 32

Wadati

This is a program to make Wadati diagrams and apparent velocity from a Nordic file with one or many events. The apparent velocity is calculated from the arrival times and the calculated epicentral distances as given in the S-file. The apparent velocity is thus approximate and affected by the location. The purpose of the program is to calculate Vp/Vs values for individual events and calculate the average for a group of events. In addition, the program can calculate the apparent velocity for each event based on P or S-times. Wadati diagrams with plot can also be calculated directly from EEV.

The information can be used to obtain a first impression of crustal parameters. For each calculation, events can be selected based on: Minimum number of stations, maximum rms of the fit (S-P vs P, or arrival times), and minimum correlation coefficient of the fit. For the apparent velocity calculation, the data can also be selected in distance and azimuth ranges.

The output gives:

- T0: Wadati calculated origin time
- N: Number of stations used for Vp/Vs
- VPS: Vp/Vs ratio
- NP: Number of stations for P- velocity
- NS: Number of stations for S-velocity
- AVSP: Average S-P times with sd
- AVDI: Average distance with sd

The average Vp/Vs is calculated for the whole data set. Individual Vp/Vs values outside the range 1.53 to 1.93 are excluded. An output file wadati.out is generated. A minimum of 3 stations is required for an event to be used. Only same type phases are used (like PG and SG).

Example of a run to calculate Vp/Vs

```
Input file name
collect.out
Wadati (1), apparent velocity (2) or both (3) ?
1
Wadati parameters: Minimum number of stations
3
Maximum rms
1
```
Minimum correlation coefficient
0.9
1994 616 1841 28.3 No data for Wadati
1994 10 4 1322 55.8 No data for Wadati
1995 822 0141 5.3 T0: 141 8.1 N: 8 VPS: 1.84 RMS: 1.19 CORR: 0.997
1995 1120 0401 58.9 T0: 41 59.5 N: 7 VPS: 1.76 RMS: 0.87 CORR: 0.995
1996 6 3 1955 35.6 No data for Wadati
1996 6 3 1955 35.6 No data for Wadati
1996 6 3 1955 40.1 No data for Wadati
1996 6 6 0648 29.8 T0: 648 1.1 N: 22 VPS: 1.51 RMS: 11.00 CORR: 0.380
1996 6 6 0648 30.6 T0: 648 38.7 N: 3 VPS: 2.16 RMS: 0.80 CORR: 1.000
1996 6 6 0648 29.8 No data for Wadati
1996 6 7 1325 28.5 T0: 1325 28.1 N: 9 VPS: 1.72 RMS: 1.06 CORR: 0.973
1996 6 7 1325 29.1 T0: 1325 28.1 N: 9 VPS: 1.72 RMS: 1.06 CORR: 0.973
1996 610 4 47.0 No data for Wadati
1996 610 1 4 47.0 No data for Wadati
1996 623 0117 57.8 No data for Wadati
1996 623 0117 58.1 No data for Wadati
1996 625 0337 31.7 T0: 337 33.1 N: 21 VPS: 1.75 RMS: 1.36 CORR: 0.999
1996 7 5 0220 46.5 T0: 220 45.8 N: 6 VPS: 1.76 RMS: 0.28 CORR: 0.999
1996 713 0556 46.0 No data for Wadati
1996 718 0946 51.4 No data for Wadati
1996 718 2255 6.0 No data for Wadati
1996 726 0742 12.0 T0: 742 11.8 N: 6 VPS: 1.74 RMS: 0.47 CORR: 0.993

Number of events for which vp/vs were calculated 9
Number of events selected for average 3

Average VP/VS = 1.75 SD= 0.01 N= 3

Example of a run to calculate apparent velocity

Input file name
collect.out
Wadati (1), apparent velocity (2) or both (3) ? 2
Apparent velocity parameters: Distance range
50 200 Azimuth range
0 180
Minimum number of stations
2
Maximum rms
1
AVSP: 30.212.7 AVDI: 152.0 30.7 NP: 4 VP : 6.95 RMS: 0.20 CORR: 1.000
AVDI: 143.3 31.1 NS: 3 VS : 3.82 RMS: 0.25 CORR: 1.000
AVSP: 23.0 9.5 AVDI: 158.7 26.6 NP: 3 VP : 6.76 RMS: 0.00 CORR: 1.000
AVDI: 90.5 35.8 NS: 6 VS : 3.72 RMS: 1.79 CORR: 0.998
<table>
<thead>
<tr>
<th>AVDI</th>
<th>AVSP</th>
<th>AVDI</th>
<th>AVSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>89.0</td>
<td>50.536.0</td>
<td>145.8</td>
<td>106.0</td>
</tr>
<tr>
<td>35.6</td>
<td>38.2</td>
<td>40.8</td>
<td>39.2</td>
</tr>
<tr>
<td>NS: 6</td>
<td>NP: 6</td>
<td>VP: 6</td>
<td>VP: 5</td>
</tr>
<tr>
<td>VS: 3.69</td>
<td>RMS: 7.60</td>
<td>RMS: 6.75</td>
<td>RMS: 6.75</td>
</tr>
<tr>
<td>CORR: 2.78</td>
<td>CORR: 0.42</td>
<td>CORR: 0.94</td>
<td>CORR: 0.97</td>
</tr>
</tbody>
</table>

Output file is *wasati.out*
Chapter 33

Calculating spectra, the SPEC program

The SPEC program is used for making spectra of many seismic signals in a semiautomatic manner. It can be used for several investigations:

A: Making a large series of signal spectra, which can be corrected for instrument and path.

Average spectra are calculated. There are two options for further processing the calculated spectra:

Option (1): Calculate acceleration density spectra which are plotted compared to the Peterson noise model.

Option (2): Using the slope of the flat part of the displacement spectra to calculate the near surface attenuation kappa (see §8.9).

B: Making relative spectra of seismic events or background noise in order to determine the soil response. When using relative spectra of horizontal versus vertical components, this is referred to as the Nakamura method [Nakamura, 1989].

C: Making relative spectra of signals from two stations in order to determine Q. The program makes output files for generating GMT plots in addition to standard SEISAN plots.

Note: Parameter file has changed between SEISAN 7.2 and 8.0 (number of windows and overlap has been added).

The program can technically operate in two ways: (1) Making relative spectra of a series of pairs of stations terminated by the average spectra, (2) Making a series of spectra for a number of stations and events. The spectra can be corrected for distance, Q, and instrument response. In addition, the spectral levels can be expressed in moment or moment magnitude calculated in the same way and with the same units as in MULPLT. All relevant parameters are taken from the CAT files, the CAL files and the input parameter file for SPEC. Window selection for the spectra can be specified to be related to the P, S arrival times or the earthquake origin time and it is thus possible to automatically make e.g. S-wave spectra of a large set of stations and events. Optionally, noise spectra, can be calculated together with the signal spectra. The noise window is selected at the start of the waveform file.

Before the program is started up, the input files must be prepared. The program need two input files. The parameter file (default spec.par) gives the parameters to use and the list of stations to process. The event file (default spec.inp) is a CAT file with events to use or a filenr.lis type file with waveform file names (can only be used if no readings are needed, like for Nakamura studies). An example of a spec.par
and spec.inp file is found in DAT. These files can be used immediately with the test data set.

The program produces several output files. The main output is in spec.out with the parameters used, the station event combinations used and error messages. The other files are giving output of most graphs shown. These ASCII output files can be used in other plotting programs, however they have been specifically formatted for the SEISAN GMTXY plotting script. Note that the numerical values given for the spectral output given in those files is the same that appear on the plots and the values are linear. So if a spectrum has been instrument corrected and smoothed, that is what is given. Similarly if a relative spectrum is used. The number of files depends on number of stations used. Examples of files could be

- spec_all_ASK_S_Z.out: All spectra from ASK, S Z
- spec_all_BER_S_Z.out: All spectra from BER, S Z
- spec_all_gmt.out: All spectra from ASK and BER
- spec_ave_ASK_S_Z.out: Average spectrum from ASK, S Z
- spec_ave_BER_S_Z.out: Average spectrum from BER, Z Z

In order to plot these files with GMTXY (only Unix), give e.g. command

```
gmtxy spec_all_ASK_S_Z.out
```

There is one more output file, spec_amps.out, which gives the log log spectra of of all traces calculated (possibly instrument corrected) before smoothing takes place.

Limitations of amount of data: The program is set up to handle 300 spectra of up to 30000 points each for one run. The dimensions can be increased in spec.for, however the program must then be recompiled. The spectral windows are 10% tapered. The analyzed signals will be checked for clipping and rejected if clipped. A message is then given in spec.out.

The spec.par file The file contains alternate lines of parameter names and parameter values, and must contain the number of lines shown in the example below.

```
selection criteria 1: P, 2: S, 3: S from P, 4: abs
2
start
1
window length, # of windows, overlap
5.0,1,1
number of times to smooth
5
gain factor of channel 1
1
noise spectrum 0: n 1: y
1
make relative spectras 1: y, 0; n
1
plot pics
1
frequency band to use
1.0 7.0
response removal: 0: none, 1: displ., 2: vel., 3: accel. 4. noise pow. 5. kappa
0
rotate: 0: no, 1: yes
0
q0, qalpha and kappa
```
-1.0,0,0
distance correction
1
minimum correlation and minimum sn for kappa
0.5 2.0
velocity and density
0.0 0.0
magnitude spectrum
0
stations and components, format a5,1x,a4,1x,a5,1x,a4

The parameters:

**Selection criteria**: Determines how the start of the time window is selected. 1: Start with the P-arrival time, 2: Start with the S-arrival time, 3: Start with the S-arrival time calculated from the P-arrival time assuming a P to S velocity ratio of 1.78, 4: Start with 'start' (see next parameter) seconds after the origin time as given in the CAT file header. This option can be used if no readings are available in the CAT file. When using a P or S-time for start of window, the program uses the first P or S phase found in the CAT file for a given station. Component is of no importance here, so there is only a need for e.g. one P-time for the station being processed if 3 component data is used. This is also the case when rotating the signal, see below. However, on the trace plots, only readings on those components shown will be seen on the plots.

**Start**: If the selection criterion is 1,2 or 3, this is the number of P or S travel times (from the origin) used to find start time of window. Use 1.0 if the window shall start exactly at the phase time picked. If selection criteria is 4, start is the number of seconds after the origin time.

**Window length, # of windows, overlap**:

- Window length: Window length in secs for both signal and noise (if selected).
- # of windows: If more than 1, spectra will be made in several windows following the first window and average spectra will be made. This option can only be used if selection criteria is 4. Used for noise studies or Nakamura studies.
- Overlap: Windows can overlap (factor < 1.0) exactly follow each other (factor=1.0) or have gaps (factor > 1.0). E.g. 0.9 is equal to 10% overlap.

**Number of times to smooth**: Number of times to smooth, 0 means no smoothing.

**Gain factor of channel 1**: Factor that the spectral level for channel 1 is multiplied with. This can be used if the response shape is the same for the two channels and only the levels are different. If the shape is also different, set factor to 1 and use response removal below.

**Noise spectrum**: If 0, no noise spectrum, if 1, make noise spectrum. The noise window is taken from the beginning of the trace and the window length is the same as given above.

**Make relative spectra**: If zero, no relative spectra, if 1, make relative spectra. The relative spectra will appear one on each page, and the average relative spectrum on the last plot (see Figure 33.1). If no relative spectra are chosen, only one trace and one spectrum is shown per page and the average spectrum is shown on the final plot. MUST BE SET to 1 to calculate Q, see below.
Plot pics: If 1, the phase pics in the CAT file spec.inp will be plotted.

Frequency band used: Lower and upper frequency bands for the spectral plots.

Response removal: If 0, no response is removed, else 1: displacement, 2: velocity, 3: acceleration (units is nm, nm/s and nm/s*s), 4: Power spectral density in dB relative to ((1m/s**2)**2)/Hz. This option is used for seismic background noise studies., 5: Determine kappa. The flat part of the spectrum (frequency below corner frequency) is approximated by a straight line and kappa calculated for each event and the average at the end (in spec.out file and on final plot). The spectrum will normally be corrected for Q, BUT NOT kappa. For more details, see [Havskov and Ottemøller] [2010]. Make sure to set appropriate frequency limits and correct distance corrections. Can be used for both P and S-spectra. A cal file for each channel must be available in the CAL directory (see section 4.6). For relative spectra, the response removal has no importance if the response is the same for the channels compared. A simple correction can be made with “Gain factor of channel 1” parameter above. NOTE: If moment or magnitude spectrum is made, response removal MUST be 1.

Rotate components: If 1, the horizontal components are rotated. This means that if the user has specified N or E, radial or transverse respectively will be used instead. The original data remain unchanged. If start time of spectra are chosen by using P or S, there must be a reading from those components if the pics are to be plotted. If the parameter is zero, no rotation is done. See also MULPLT for more details of rotation.

Q0, qalpha and kappa:

Q-correction: Parameters in Q-relation Q = Q0**qalpha used for spectral correction (see also section on MULPLT for standard attenuation relations). Only used if response is removed. If first 2 parameters are 0,0, no Q-correction. New from SEISAN7.2 is that a kappa correction also can be used (see MULPLT spectral section). Note that from version 10.1, Q=Q0 for f ¡ 1 Hz.

Calculation of Q:
If Q0 and qalpha is set to -1,0, the relative spectra will be used to calculate q as a function of f (see standard relations in MULPLT section) and the plots will show q as a function of f. This can be used for both P-waves and S-waves. The distance correction MUST be set, S-velocity must be given (see below) and it is recommended to assume body wave spreading (amplitude proportional to 1/distance, factor is 1.0 below). If the response of the 2 stations is not identical, correction for response must also be made. There must be an origin time and phase readings for components used must be available in order to calculate Q. Q is calculated as pi * f * (t2-t1) / (ln(A2(f)/A1(f)) + alpha * ln(t2/t2)), where A1 and A2 are spectral levels at frequency f for the two stations, t1 and t2 are travel times and alpha is geometrical spreading exponent (1.0 is body wave spreading). Q values lower than 1 and higher than 5000 are not used, the Q=f plot might then display a long straight line. The Q=f plot is calculated from the ‘good’ values.

Distance correction alpha: The spectral amplitudes are multiplied by R**(distance correction) if different from zero. This option MUST be set if moment or moment magnitude options (see below) are selected as well as calculation of Q. However, it can be used without instrumental correction. For body waves, use 1. Note that the geometrical spreading use here is simpler than used in MULPLT.

Minimum correlation coefficient and minimum signal to noise ratio for kappa: The minimum correlation coefficient and signal to noise ratio for an event to be included in average kappa. The coefficients are from the linear fit to the flat part of the spectrum.

Velocity and density: Velocity (km/sec) and density (g/cm*cm) used for calculating moment spectra. If set to 0,0, no moment spectra are calculated. See section on MULPLT for details of calculation.
**Magnitude spectrum:** If 1, the spectral level is converted to moment magnitude, see MULPLT for details of calculation.

**Stations and components:** Station-component pairs used, one pair per line, format (a5,1x,a4,1x,a5,1x,a4). If no relative spectrum is used, the first station-component on the line is used.

Averaging in spec:
Q: For each frequency, the average linear 1/Q and corresponding sd is calculated. The upper and lower bounds are calculated by subtracting and adding the sd. These values are then converted back to Q and finally the log is taken. Only the ‘good’ individual values are used. There is a possibility that the lower bound becomes negative. In that case, the log Q is set to zero. Because the average is made in 1/Q, the upper and lower bounds curves will not be symmetric around the average Q-curve.

Power spectrum: For each frequency the dB values are averaged and upper and lower curves should be symmetric.

Kappa: Same as for Power spectrum.

Other spectra: The linear spectra or relative spectra are averaged. The sd used in the log spectra are calculated by subtracting log average spectrum from log(average spectrum + sd).

Running the program:
The program gets the first pair of stations (or one station) from spec.par, calculates the spectra using the list of events in spec.inp and at the end of the station list, calculates the average spectral ratios for all pairs (max 100). All spectra are then shown on one plot together with averages and standard deviation. Then the next pair of stations is processed in the same way and the program continues until the end of file spec.par. Each pair of stations with signals and spectra is plotted on one page. If no relative spectra are made, the plots look similar except that only one station is shown. Hard copy plots are made for each page and sent to the printer if specified (see below). The hard copy postscript file is called spec.eps when the program finishes, a file with the last plot is available on the disk. For each spectrum (relative or single), the average spectrum (or Q) is calculated both as an average of the log spectrum and as an average of the linear spectrum. There is no frequency weighting and since all values shown on the plot are used, the average value will be more representative of the high frequency part of the spectrum since there are more values. This can be regulated by choosing another frequency range. The average spectra shown on the last plot are log-averages. If option to calculate Q is used, the plots show 1/Q as a function of frequency instead of relative spectra (proportional to relative spectra). For each event, Q0 and qalpha are calculated.

When calculating kappa, the average spectrum do not have much physical meaning since the averages are made from absolute spectra of events that might have very different moments. So the kappa calculated from the average spectrum is not to be used.

Interactive output of level and frequency: With a spectral ratio (or Q) plot on the screen, position the cursor at the point of interest on the spectrum and click. The level and frequency will now be displayed on the right side of the plot.

The output file spec.out gives details of the run like averages and missing data. The output file spec.ave.out gives the x and y-values of the average spectrum IF IT HAS BEEN PLOTTED ON THE SCREEN. File spec.rel.out gives the values of the relative spectra.

There are 4 interactive input options:
0: All spectra are calculated but not sent to the plotter or screen except the last plot with the average spectra (sent to both screen and printer). Used for checking the files or making a final run. If no relative spectrum is chosen, no final plot is made. For each station and event combination, check lines are written
out on the screen.

1: All plots are shown on the screen, but not sent to the laser printer.

2: All plots are shown on the screen and at the same time sent to the laser printer.

3: No plots are shown on the screen, all are sent to the laser printer. For each station event combination, check lines are written to the screen.

4: Only final plot on laser.

5: No plotting so no graphics window come up anytime. Can be used e.g. for batch mode.

How to run the program with only waveform files available: Two options:

(A) Using S-files
Step 1: Generate S-files in your local directory with AUTOREG,
Step 2: Make the spec.inp file with COLLECT.

(B) Using filenr.lis
Step 1: Make a dirf of waveform files to use
Then use filenr.lis as input file name

With only waveform files and no readings in the spec.inp file, it is only possible to use option 4 (absolute time) for start criteria. Since the events have not been located, the "origin time" read from the S-files will be identical to the waveform file start time, so the parameter "start" can then be set to number of seconds after waveform file start time. Figure 33.1 shows an example.
Figure 33.1: An example of using the SPEC program. On top the original traces are shown with windows chosen, in the middle the spectra of each channel and at the bottom, the relative spectrum. Lower right shows the input parameters used. In some cases (kappa and Q) the values calculated for this case are also shown.
Figure 33.2: An example of a GMT plot. The figure shows an example of making noise spectra of several traces.
Chapter 34

Seismic risk related programs

This section is written by M. B. Sørensen, based on a previous version by K. Atakan. Extensive testing of the programs was done over the years by many users.

Introduction

Currently, the SEISAN package includes a series of stand-alone programs and Seisan Explorer (SE) functions that can be used in a number of tasks that are needed to perform seismic hazard analysis. The basic requirements for performing a probabilistic seismic hazard analysis (PSHA) may be summarized as follows:

- Collect the best possible earthquake catalog for the study area.
- Homogenize the earthquake catalogue and assess the completeness.
- Define the seismic source zones.
- Prepare input parameters from the earthquake catalogue for each source zone.
- Define ground motion prediction equations (GMPE) for the region.
- Compute hazard in terms of different ground motion parameters (e.g., peak ground acceleration (PGA), peak ground velocity (PGV) or spectral acceleration (SA) at different periods).
- Assess site effects.
- Prepare response spectra.

34.1 useful programs for hazard

Following is a list of programs that may be useful during a PSHA study. Most of these programs are described in more detail in different sections of the SEISAN manual.

SELECT: Select a subset of earthquake data according to given criteria. This can be useful both in connection with initial quality testing of earthquake catalogs and for selecting events within a given source zone. SE has a similar but more advanced function.

CATSTAT: Program to compute and plot the yearly, monthly and daily number of events in addition
to the time-of-day distribution of events from a given catalogue.

**NORHEAD:** The program can move magnitudes form following header lines to empty space on first header line.

**CAT AGA:** Program to reorder the hypocenter lines in a CAT-file according to hypocenter agency, in order to put the prime estimate in the beginning.

**CLUSTER:** Program that searches for dependent events, defined through time, distance and depth windows, in a given CAT file.

**ASSO:** Program to merge events which are close in time, magnitude and space in a database. This program is useful when merging catalogs to avoid double entries. ASSOCI will perform a similar search, but only based on the time difference between events. Furthermore, ASSOCI can only merge two events at a time. It is thus recommended to use ASSO rather than ASSOCI when merging catalogs for hazard assessment purposes.

**EXFILTER:** Identifies probable explosions, based on user-defined parameters involving time-of-day distribution and the mining locations. It can be used for catalogue clean-up and for discrimination between earthquakes and man-made explosions.

**MAG:** Magnitude regression and conversion program. Prepares a plot showing the data scatter and the best-fitted line for conversion between two magnitude types. Magnitude conversions can then be performed after a user defined priority list.

**EPIMAP:** Plots coastlines, national boundaries and earthquake epicenters. It is also possible to select a subset of earthquakes from a chosen polygon on the epicenter map.

**BVALUE:** Prepares magnitude-frequency of occurrence diagrams and computes a- and b-values with maximum likelihood and least square approximation. In addition, the threshold magnitude and the maximum observed magnitude can be obtained. It is generally recommended to use the Gutenberg-Richter or Weichert functions built into Seisan Explorer instead of BVALUE.

**SPEC:** Computes amplitude spectra for a given set of earthquake records and plots spectral ratios. It can be used to assess local site effects.

**CRISIS2012:** Computes seismic hazard in terms of the probability of exceedance vs. earthquake intensity measure. Any intensity measure can be provided through user defined ground motion prediction equations in addition to a large number of built-in relations. Epistemic uncertainty can be considered through the definition of a logic tree. Results can be provided as hazard maps or in terms of hazard curves for selected sites. The program can furthermore provide a deaggregation of the results. The program only runs under Windows and needs to be installed separately. Look for a MSI file in SUP.

**SEISAN EXPLORER:** A number of functions have been built into Seisan Explorer to aid the preparation of input to a PSHA:

- Function 'Gutenberg-Richter relation' allows for determining a- and b-values for a given catalog using different magnitude intervals and bin sizes (similar to the BVALUE program).
- Function ‘Poisson distribution’ allows for visually checking whether an earthquake catalog is Poisson distributed or not.
- Function ‘Completeness check’ provides a ‘staircase plot’ of the earthquake catalog, showing for different magnitude classes the cumulative number of events as a function of time. This plot allows for defining catalog completeness for different magnitude classes.
- Function ‘Weichert method’ allows for determining a- and b-values for a given catalog, accounting for varying catalog completeness for different magnitude classes.

Probabilistic seismic hazard computations are done using the Crisis2012 program. In addition, the programs listed above and a number of other programs that manipulate earthquake data within the SEISAN package are useful tools to assess the parameters that are needed to perform a seismic hazard analysis for an area of interest. Crisis2012 is developed by Mario Ordaz of the Institute of Engineering, UNAM [Ordaz 1991, 1999].

**Step by step procedure for seismic hazard analysis**
34.1. USEFUL PROGRAMS FOR HAZARD

Following is a summary of the steps that need to be completed in order to produce a seismic hazard map.

1. Compile a catalogue for the area of interest from local, regional and global sources. Most commonly, several catalogs are merged. After quality control of the individual catalogs (SELECT, CATSTAT and some of the Seisan Explorer tools can be useful for this) the catalogs can be merged using SPLIT and ASSO.

2. Evaluate the preliminary catalogue completeness by using the 'Catalog completeness' function of Seisan Explorer.

3. Convert magnitudes into one uniform magnitude, preferably to moment magnitude MW. To do this, regression curves must be prepared for different magnitude scales. Program MAG can be used for this purpose.

4. Clean up the catalogue for dependent events (i.e. induced seismicity, non-earthquakes, foreshocks, aftershocks, earthquake swarms). Here a search has to be made for clusters of events both in time and space. Program CLUSTER can be used for this purpose. The probable explosions may be removed by using the program EXFILTER. Following, the 'Poisson distribution' function of Seisan Explorer can be used to confirm that the cleaned catalog is Poisson distributed.

5. The evaluation of the catalogue completeness is dependent upon the clean-up process and the magnitude unification. It is therefore necessary that steps 2-4 be repeated until a reliable catalogue is prepared.

6. Delineate earthquake source zones for the area of interest. The zonation can be based on a seismicity map with the clean catalog and additional information from geology, geophysics, seismotectonics, paleoseismology etc. A seismicity map can be prepared using EPIMAP or GMAP (with Google Earth). Both programs have zooming and area selection procedures which can be used to define the source zone polygons.

7. For each earthquake source zone select the subset of events that fall in the chosen area. This can be done by using the SELECT program. Alternatively, EPIMAP can be used to draw polygons interactively on the screen and put the subset of events within this polygon into a file.

8. The seismicity within each source zone is assumed to be uniform following a Poisson distribution. This can be checked using the 'Poisson distribution' function of Seisan Explorer. For each source zone, the following parameters then need to be defined:
   - The a and b-value of the Gutenberg-Richter relation: these can be defined using the 'Weichert method' or 'Gutenberg-Richter relation' functions of Seisan Explorer or from BVALUE. When applying these programs, magnitude interval and bin sizes must be chosen critically, taking into account the catalogue completeness and the detection threshold.
   - Maximum expected magnitude with its standard deviation: This is usually inferred through other available information, such as geology, palaeoseismicity, or subjective judgment of the scientist.
   - Maximum observed magnitude: This is the largest magnitude observed within the catalogue time span.
   - Threshold magnitude: The so-called lower bound magnitude, which is chosen, based on engineering considerations. Usually magnitudes less than 4.0 are not considered engineering significant.

9. It should be assessed whether there are characteristic earthquakes in the region of interest. This can be done through careful examination of the catalogue and the active faults in the area. If characteristic earthquake sources are included in the analysis, it is important that the seismicity rate is reduced correspondingly in any area sources overlapping with the characteristic fault sources.
10. Ground motion prediction equations (attenuation relations), describing the level of ground shaking as a function of magnitude and distance, must be defined for each source zone. Such relations are based on empirical ground motion observations. If no suitable relations are available for the chosen study area and cannot be derived based on existing data, relations can be adopted from tectonically similar regions. In that case it is important to check the predictions of the chosen relations against available recordings from the study area. Many relations are defined for different types of ground conditions (rock, stiff soil, soft soil) and it is important that appropriate relations are chosen.

11. Run the CRISIS2012 program in order to set up a hazard model with the parameters determined in steps 6-10. The model can be defined through the graphical user interface of CRISIS2012. Optionally, a logic tree can be set up to account for epistemic uncertainty. Results can be displayed as hazard maps or hazard curves directly, or exported as ASCII files. There is also an option for deaggregating the results.

12. The local site conditions should be considered for critical sites. SPEC program can be used to obtain the amplification factors due to unconsolidated sediments. These factors can be used later to adjust the response spectra.

Many of the programs mentioned above are described individually throughout this manual at different sections. In the following the programs that are directly relevant to hazard computations and not described in other sections of the manual are explained in detail.

34.2 CRISIS2012

CRISIS2012 is a computer program to compute seismic hazard in extended regions. It was developed at the Institute of Engineering, UNAM, Mexico, by Mario Ordaz (mors@pumas.ingen.unam.mx), Armando Aguilar and Jorge Arboleda.

Basic input data are: geometry of the sources, seismicity parameters of the sources, and ground motion prediction equations. In addition, the spectral ordinates, for which the hazard is calculated, must be defined together with the calculation grid and parameters related to the integration and display of results. Details about the use of CRISIS2012 are given in the separate CRISIS2012 manual PDF file contained in the MSI installation file in SUP.

CRISIS2012 is only available under Windows and needs to be installed separately. An MSI file is available in SUP. When opening the MSI file, an installation widow appears asking a number of questions in Spanish. The following choices should lead to successful installation of CRISIS2012:

1st window: Siguiente (next)

2nd window: Choose the location where CRISIS2012 should be installed (browse with 'Examinar'). Choose whether CRISIS2012 should be available for all users (Para todos los usuarios) or current user only (Solo para este usuario) Siguiente (next)

3rd window: Siguiente (next)

4th window: Cerrar (close)
34.3 SEISAN EXPLORER:

During the process, it may be necessary to update the .NET framework of the computer. After installation, a number of files will be available at the location specified in the 2nd installation window. CRISIS2012 is launched through CRISIS 2012.exe. A comprehensive manual is found in CRISIS 2008 Manual.pdf and through the help function of CRISIS2012.

34.3 SEISAN EXPLORER:

The Seisan Explorer functions which are specifically relevant for seismic hazard assessment are described in the following. The user is referred to the separate section on Seisan Explorer for details on the general use of the program.

**Function 'Gutenberg-Richter relation'** allows for determining a- and b-values of the Gutenberg-Richter relation for a given catalog using different magnitude intervals and bin sizes. The Gutenberg-Richter relation states that

\[
\log(N) = a - bM
\]  

(34.1)

where log is the base-10 logarithm and N is the number of events with magnitude M (or larger in case of regression on cumulative data, see below).

The user can choose which magnitude type to use (M is the first magnitude given for each event). The database is read, and a histogram is plotted showing the number of events in different magnitude bins. The minimum magnitude to be considered and the bin size can be chosen by the user. Optionally, the incremental number of events and the cumulative number of events above a given magnitude can be overlain as symbols. The a- and b-values can be determined by regression on either the incremental or the cumulative values. The starting magnitude is selected by the user and should be set such that only the complete part of the catalog is considered. It is also possible to set a maximum magnitude in the regression. The obtained a- and b-values are returned, and the fit is compared to the data in the plot.

**Function 'Poisson distribution'** allows for visually checking whether an earthquake catalog is Poisson distributed. The function reads the database and plots a histogram showing the number of 1-year intervals with a given number of earthquakes, as a function of the annual number of earthquakes. This histogram is compared to the theoretical Poisson distribution derived from

\[
P(N = n) = \frac{\nu^n}{n!} e^{-\nu},
\]

where n is the number of earthquakes for a given year and \( \nu \) is the mean annual number of events. A good fit between the histogram and the theoretical curve indicates that the data fulfills the Poisson distribution.

**Function 'Completeness check'** allows for defining catalog completeness for different magnitude classes. The function provides a 'staircase plot' of the earthquake catalog, showing for different magnitude classes the cumulative number of events as a function of time. Assuming that the catalog is complete during the most recent part of the time interval, the user can search for a change in slope, indicating a change in seismicity rate. Such change is interpreted as a change in catalog completeness. The completeness time can thus be read as the time where the staircase plot changes slope for a given magnitude class and entered into a table. The plot can be zoomed by scrolling and moved by dragging with the mouse. The user can select the minimum magnitude to be considered and the magnitude interval for each curve.
The entered completeness values are shown in the plot and saved to a file se-completeness.out, to be read by function ‘Weichert method’, when pushing ‘Replot’.

**Function ‘Weichert method’** allows for determining a- and b-values for a given catalog, accounting for varying catalog completeness for different magnitude classes following [Weichert 1980]. The magnitude classes and their corresponding completeness times are read from the file se-completeness.out, which is generated by function ‘Completeness check’. The user can choose which magnitude classes to consider through ‘Select intervals to be used’. The function performs a standard linear regression for the Gutenberg-Richter a- and b-values for the complete part of the catalog. In addition, a regression is performed following [Weichert 1980]. This regression returns the b-value, the Mmin value used and a value of Ny(M) for a magnitude selected by the user, defined through:

\[
\log(Ny) = \log(Ny(M_{\text{min}})) + b(M_{\text{min}} - M),
\]

(34.2)

where log is the base-10 logarithm and Ny(M) is the annual number of earthquakes with magnitude M.

### 34.4 CLUSTER:

The purpose of CLUSTER is to find and remove dependent events (foreshocks, aftershocks and swarm events) from a catalog. In contrast to the old CLUSER in SEISAN before version 10.0, this completely new version will search for dependent events using more parameters: Time difference, epicentral distance and depth distance. These parameters can be magnitude dependent and can be different before and after the main event. The program requires an input parameter file cluster.def which can be placed in working directory or DAT. An example is given in DAT and seen below.

For declaring an event as a dependent event, the dependent event must occur at less than a given time difference from the main event, it must be smaller than a specified magnitude (relative to the main event), the epicentral or hypocentral distance between main and dependent event must be less than a given value and the difference in hypocentral depth between the two events must be less than a given value. Hypocentral distance is calculated as the direct distance between main shock and dependent event. The time and distance limits are generally magnitude dependent so larger values are used for larger events. Different magnitude dependence can be used before and after the main event.

If the event has no location or magnitude, it will not be considered as a possible dependent event and will be included in the list of main shocks.

The magnitude used is selected among all prime magnitudes in the file (the 6 magnitudes on main header line, can be 2 lines if more than 3 magnitudes). The magnitudes can be given an order of priority in SEISAN.DEF:

<table>
<thead>
<tr>
<th>KEYWORD............</th>
<th>Comments.............</th>
<th>Par 1</th>
<th>Par 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAGNITUDE_ORDER</td>
<td>LBER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAGNITUDE_ORDER</td>
<td>LNAO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In this example, LBER (local magnitude calculated by BER) is first chosen, if not there, LNAO and if not there either, the first magnitude found in file. If no order is given in SEISAN.DEF, the magnitude used will be the first found for the event, irrespective of type or agency. It is possible to leave either magnitude or agency blank in which case the blank represents a wildcard.

Potentially all events can be a mainshock so all combinations, within the time limit, will be searched. A search is first made after the main event (for aftershocks) and then before the main event (for foreshocks).
34.4. CLUSTER:

Input:
A file in Nordic format. If only header lines, there must be a blank line between events.

Output:
All dependent events will be removed and put out in a file called cluster_reject.out while the cleaned catalog will be in file cluster_use.out. Optionally, a debug output can be made, either on screen or in a file (see cluster.def). An example is seen below

<table>
<thead>
<tr>
<th>Main</th>
<th>2012</th>
<th>210</th>
<th>1222</th>
<th>22.3</th>
<th>66.4</th>
<th>14.2</th>
<th>14.0</th>
<th>0.9</th>
<th>LTES</th>
</tr>
</thead>
<tbody>
<tr>
<td>After</td>
<td>2012</td>
<td>212</td>
<td>2332</td>
<td>7.8</td>
<td>71.2</td>
<td>-8.1</td>
<td>13.0</td>
<td>3.5</td>
<td>LTES</td>
</tr>
<tr>
<td>After</td>
<td>2012</td>
<td>213</td>
<td>853</td>
<td>34.5</td>
<td>71.0</td>
<td>-8.0</td>
<td>19.7</td>
<td>1.3</td>
<td>LTES</td>
</tr>
<tr>
<td>After</td>
<td>2012</td>
<td>216</td>
<td>1322</td>
<td>6.1</td>
<td>71.1</td>
<td>-7.7</td>
<td>0.1</td>
<td>1.5</td>
<td>LTES</td>
</tr>
<tr>
<td>Fore</td>
<td>2012</td>
<td>124</td>
<td>027</td>
<td>10.9</td>
<td>71.1</td>
<td>-8.0</td>
<td>43.2</td>
<td>2.3</td>
<td>LTES</td>
</tr>
<tr>
<td>Fore</td>
<td>2012</td>
<td>122</td>
<td>1248</td>
<td>16.6</td>
<td>71.1</td>
<td>-8.3</td>
<td>23.7</td>
<td>1.7</td>
<td>LTES</td>
</tr>
<tr>
<td>Fore</td>
<td>2012</td>
<td>120</td>
<td>128</td>
<td>10.3</td>
<td>71.1</td>
<td>-7.5</td>
<td>13.2</td>
<td>1.8</td>
<td>LTES</td>
</tr>
<tr>
<td>Main</td>
<td>2012</td>
<td>210</td>
<td>1543</td>
<td>54.2</td>
<td>73.3</td>
<td>7.8</td>
<td>15.0</td>
<td>2.8</td>
<td>LTES</td>
</tr>
<tr>
<td>Main</td>
<td>2012</td>
<td>210</td>
<td>1549</td>
<td>52.9</td>
<td>-27.6-176.6</td>
<td>33.0</td>
<td>5.0</td>
<td>BNAO</td>
<td></td>
</tr>
<tr>
<td>Main</td>
<td>2012</td>
<td>210</td>
<td>1654</td>
<td>31.8</td>
<td>2.2</td>
<td>122.2</td>
<td>33.0</td>
<td>5.4</td>
<td>BNAO</td>
</tr>
</tbody>
</table>

Each potential main (Main) event is listed. To be considered a potential main event, the event must have both location and magnitude. If a dependent event is found, its type is shown (After or Fore) with the parameters: M: The interpolated main shock magnitude used, T: The interpolated maximum time difference used and the actual time difference (days), D: The interpolated maximum distance (km) used and the actual distance. In the above example, the upper magnitude limit for dependent events is 2.5, main event is 3.5, the limit in distance is 26 km and in time 22 days. Directly following the Main event comes a listing of all dependent events. The next potential main event will then be the first unused event following the previous Main event and that Main event can then jump back in time since there might be several events following the previous Main event which were not declared as dependent events.

Example of a cluster.def file

This file is parameter file for cluster. The file can be in working directory or DAT.
Only the lines with recognized keywords and a non blank field under Par 1 will be read. The comments have no importance. Lines with keywords MAGS AFTER DIST TIME and MAGS BEFORE DIST TIME must come grouped together in increasing magnitude order. Parameters for these lines are:
Reference magnitude of main event, maximum magnitude of corresponding after/foreshock (AFTER/FORE), distance(km) and time(days).
All number must be real numbers. Columns Par 1-Par 4 start in columns 41, 51, 61 and 71.
All keywords in capital letters.
Hypocentral distance is calculated as the direct distance between main shock and dependent event.
Parameter values are interpolated for magnitudes between the ones given. The values shown in debug output are the interpolated values.
If event is larger than largest event in table, the values for the largest event is
used. If event is smaller than smallest event in table, no dependent events will be searched for.

The MAGS AFTER DIST TIME are magnitude of main shock, corresponding magnitude of largest aftershock, distance and time. Similarly for the foreshocks.

The MAX DEPT DIFFERENCE will consider only events with a depth difference less than MAX DEPT DIFFERENCE as a possible dependent event.

KEYWORD............Comments.............Par 1.....Par 2.....Par 3.....Par4.....
HYPOCENTRAL DIST 1.0 yes, 0.0 no 0.0
MAX DEPTH DIFF Difference in depth 50.0
DEBUG OUT 0.0 none, 1.0 screen 2.0 file 1.0

<table>
<thead>
<tr>
<th>MAGS AFTER DIST TIME</th>
<th>Mag. dep. par.</th>
<th>2.5</th>
<th>1.5</th>
<th>19.5</th>
<th>6.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAGS AFTER DIST TIME</td>
<td>Mag. dep. par.</td>
<td>3.0</td>
<td>2.0</td>
<td>22.5</td>
<td>11.5</td>
</tr>
<tr>
<td>MAGS AFTER DIST TIME</td>
<td>Mag. dep. par.</td>
<td>3.5</td>
<td>2.5</td>
<td>26.0</td>
<td>22.0</td>
</tr>
<tr>
<td>MAGS AFTER DIST TIME</td>
<td>Mag. dep. par.</td>
<td>4.0</td>
<td>3.0</td>
<td>30.0</td>
<td>42.0</td>
</tr>
<tr>
<td>MAGS AFTER DIST TIME</td>
<td>Mag. dep. par.</td>
<td>4.5</td>
<td>3.5</td>
<td>35.0</td>
<td>83.0</td>
</tr>
<tr>
<td>MAGS AFTER DIST TIME</td>
<td>Mag. dep. par.</td>
<td>5.0</td>
<td>4.0</td>
<td>40.0</td>
<td>155.0</td>
</tr>
<tr>
<td>MAGS AFTER DIST TIME</td>
<td>Mag. dep. par.</td>
<td>5.5</td>
<td>4.5</td>
<td>47.0</td>
<td>290.0</td>
</tr>
<tr>
<td>MAGS AFTER DIST TIME</td>
<td>Mag. dep. par.</td>
<td>6.0</td>
<td>5.0</td>
<td>54.0</td>
<td>510.0</td>
</tr>
<tr>
<td>MAGS AFTER DIST TIME</td>
<td>Mag. dep. par.</td>
<td>6.5</td>
<td>5.5</td>
<td>61.0</td>
<td>790.0</td>
</tr>
<tr>
<td>MAGS AFTER DIST TIME</td>
<td>Mag. dep. par.</td>
<td>7.0</td>
<td>6.0</td>
<td>70.0</td>
<td>915.0</td>
</tr>
<tr>
<td>MAGS AFTER DIST TIME</td>
<td>Mag. dep. par.</td>
<td>7.5</td>
<td>6.5</td>
<td>81.0</td>
<td>960.0</td>
</tr>
<tr>
<td>MAGS AFTER DIST TIME</td>
<td>Mag. dep. par.</td>
<td>8.0</td>
<td>7.0</td>
<td>94.0</td>
<td>985.0</td>
</tr>
<tr>
<td>MAGS AFTER DIST TIME</td>
<td>Mag. dep. par.</td>
<td>8.5</td>
<td>7.5</td>
<td>110.0</td>
<td>1000.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MAGS BEFORE DIST TIME</th>
<th>Mag. dep. par.</th>
<th>2.5</th>
<th>1.5</th>
<th>19.5</th>
<th>6.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAGS BEFORE DIST TIME</td>
<td>Mag. dep. par.</td>
<td>3.0</td>
<td>2.0</td>
<td>22.5</td>
<td>11.5</td>
</tr>
<tr>
<td>MAGS BEFORE DIST TIME</td>
<td>Mag. dep. par.</td>
<td>3.5</td>
<td>2.5</td>
<td>26.0</td>
<td>22.0</td>
</tr>
<tr>
<td>MAGS BEFORE DIST TIME</td>
<td>Mag. dep. par.</td>
<td>4.0</td>
<td>3.0</td>
<td>30.0</td>
<td>42.0</td>
</tr>
<tr>
<td>MAGS BEFORE DIST TIME</td>
<td>Mag. dep. par.</td>
<td>4.5</td>
<td>3.5</td>
<td>35.0</td>
<td>83.0</td>
</tr>
<tr>
<td>MAGS BEFORE DIST TIME</td>
<td>Mag. dep. par.</td>
<td>5.0</td>
<td>4.0</td>
<td>40.0</td>
<td>155.0</td>
</tr>
<tr>
<td>MAGS BEFORE DIST TIME</td>
<td>Mag. dep. par.</td>
<td>5.5</td>
<td>4.5</td>
<td>47.0</td>
<td>290.0</td>
</tr>
<tr>
<td>MAGS BEFORE DIST TIME</td>
<td>Mag. dep. par.</td>
<td>6.0</td>
<td>5.0</td>
<td>54.0</td>
<td>510.0</td>
</tr>
<tr>
<td>MAGS BEFORE DIST TIME</td>
<td>Mag. dep. par.</td>
<td>6.5</td>
<td>5.5</td>
<td>61.0</td>
<td>790.0</td>
</tr>
<tr>
<td>MAGS BEFORE DIST TIME</td>
<td>Mag. dep. par.</td>
<td>7.0</td>
<td>6.0</td>
<td>70.0</td>
<td>915.0</td>
</tr>
<tr>
<td>MAGS BEFORE DIST TIME</td>
<td>Mag. dep. par.</td>
<td>7.5</td>
<td>6.5</td>
<td>81.0</td>
<td>960.0</td>
</tr>
<tr>
<td>MAGS BEFORE DIST TIME</td>
<td>Mag. dep. par.</td>
<td>8.0</td>
<td>7.0</td>
<td>94.0</td>
<td>985.0</td>
</tr>
<tr>
<td>MAGS BEFORE DIST TIME</td>
<td>Mag. dep. par.</td>
<td>8.5</td>
<td>7.5</td>
<td>110.0</td>
<td>1000.0</td>
</tr>
</tbody>
</table>

There must be at least one "MAGS AFTER DIST TIME" line but "MAGS BEFORE DIST TIME" are not needed. In that case there will be no search for foreshocks.
Chapter 35

Magnitude relations, MAG

The MAG program calculates simple magnitude relations. The program has three functions: (1) Calculate parameters for a magnitude scale (Ml or Mc), (2) Calculate relation between two different magnitudes and/or spectral parameters and (3) Calculate a new magnitude as a function of an existing magnitude, a natural step following function (2). All three functions can be done at the same time. Function (3) can also be used for moving a particular magnitude type and/or agency to the first magnitude position in line 1 to be plotted with EPIMAP.

ALL HEADER LINES ARE SEARCHED FOR MAGNITUDE INFORMATION

Input: The data input is a CAT-file like one made with SELECT or COLLECT or it can be a compact file if only magnitude comparison is made. Optionally there can be a parameter file, which MUST be, called mag.par and MUST reside in the working directory. An example of the parameter file is found in DAT and also shown below. The parameter file is not needed for all operations, see details below.

1: Magnitude scales

Coda magnitude Mc: The coda magnitude scale used is

\[ Mc = A \times \log(coda) + B \times dist + C \]

where Mc is the coda magnitude, coda is the coda length in secs, dist is the hypocentral distance in km (calculated from epicentral distance and depth in CAT file) and A, B and C are constants to be determined. This is done in two ways

3d regression

\[ m = A \times \log(coda) + B \times dist + C \]

2d regression

\[ m = A \times (\log(coda) + dist_\text{coff} \times dist) + C \]

with B = A * dist_\text{coff} where dist_\text{coff} is given in the parameter file and m is the reference magnitude. SO B AND dist_\text{coff} ARE DIFFERENT. The CAT-file must contain coda readings, epicentral distances and a magnitude in the header line. A linear regression is then made between the known magnitude from
a given agency and the observed coda lengths following the relations above. The user has the option to 
choose the type of magnitude to use in the regression. Usually Ml or Mb are used. All station-event 
combinations are used to determine simultaneously the 3 constants A, B and C. Since the data often 
is too bad to determine all 3 parameters at the same time, the program will also calculate just A and 
C using a fixed user supplied value for the distance correction to the coda. The constant dist_coff is 
given in the mag.par file as the second parameter under MAG_TYP_COF (see below). IN ORDER FOR 
THE CODA SCALE OPTION TO WORK, THE DISTANCE COEFFICIENT MUST BE DIFFERENT 
FROM ZERO.

Output: On the screen the constants will be printed out and a file mag_coda.out will contain pairs of 
values m and (log(coda) + dist_coff*dist), which can be used to plot the distance corrected coda relation. 
If results from the 3D is to be plotted, dist_coff must be calculated as dist_coff=B/A, put into mag.par 
and mag run again. On the other hand, if a best dist_coff has been found, B is calculated as B=A*dist_cod

A typical coda magnitude relation is:

\[ Mc = 2.0 \times \log(coda) + 0.0035 \times dist - 0.87 \]

[Lee et al., 1972]

Local magnitude Ml:
The local magnitude scale is calculated by determining an amplitude attenuation scale using amplitudes 
and distances in CAT file. The parameters in the Ml magnitude scale are computed for every event 
individually, parameters are determined as averages of all events.

For each event (only type L and R are used) a,b,c are calculated if at least 3 stations are available using 
least squares regression as follows:

\[ \log(amp) = a \times \log(dist) + b \times dist + c \]

The relation above can be derived from the standard geometrical spreading and attenuation relations:

\[ amp = (dist \times a) \times \exp(pi \times f \times dist/(v \times q)) \]

where f is the frequency, v is the velocity and q = q0 \times f \times qalpha. The relation can be rewritten

\[ \log(amp) = a \times \log(dist) + (pi \times f \times dist)/(v \times q0 \times f \times qalpha \times 2.3) \]

Since qalpha often is close to 1.0, the relation can be simplified to the frequency independent relation:

\[ \log(amp) = a \times \log(dist) + (pi \times dist)/(v \times q0 \times 2.3) \]

If body wave spreading is assumed (a = 1), q0 = 100 and v = 3.5km/sec, the relation is

\[ \log(amp) = 1.0 \times \log(dist) + 0.004 \times dist \]

which is comparable to the relation shown below for California.

Similarly to the coda relation, a 2D relation is also calculated
\[ \log(\text{amp}) - b \times \text{dist} = a \times \log(\text{dist}) \]

where \( b = \text{dist}\_\text{coff} \) is fixed to the value given in the \text{mag.par} file (same parameter as used for coda). This gives a more stable solution, however \( b = \text{dist}\_\text{coff} \) must be determined by trial and error or fixed using known values from e.g. q-studies.

The amplitudes are assumed to be ground displacements (in SEISAN they are ground displacements highpass filtered at 1.25 Hz to resemble Wood Anderson seismograms, see MULPLT). The distance ratio between stations with the maximum distance and minimum distance must be more than 3 for the event to be selected for analysis. It is assumed that \( a \) and \( b \) will be the same for all events, while \( c \) is different (magnitude dependent). At the end, the average constants \( a \) and \( b \) are calculated of all values \( a \) and \( b \) which are not deviating too much (\( a \) must be in the range 0 to -5, hardwired). Distance attenuation coefficients \( a \) and \( b \) are supposed to be negative since amplitude decrease with distance. To get the local magnitude scale

\[ M_l = \log(\text{amp}) - a \times \log(\text{dist}) - b \times \text{dist} - C \]

the constant \( C \) must be determined by fixing the magnitude at some reference distance like the original Wood Anderson definition with \( M_l = 3 \) at \( \text{dist} = 100 \text{km} \) and \( \text{amp} = 1/2200 \text{mm} = 454 \text{nm} \) (assuming gain of the Wood Anderson seismograph to be 2080, [Hutton and Boore, 1987]). The determination of \( a \) and \( b \) does not work well unless the observations are very good. The relation for California is [Hutton and Boore, 1987]

\[ M_l = \log(\text{amp}) + 1.1 \times \log(\text{dist}) + 0.00189 \times \text{dist} - 2.09 \]

Output: On the screen the constants will be printed out and a file \text{mag.amp.out} will contain the values of \( a \), \( b \) and \( c \).

2: Magnitude relations and/or spectral parameter relations

Linear regression (maximum likelihood) can be made between any two magnitudes and/or spectral parameters on any of the header lines of an event in a CAT-file or a compact file. The user is interactively prompted for the magnitude type and/or spectral parameters and agencies to compare. If none is given, no magnitude comparison will be made. If several magnitudes/spectral parameters fit the requirement, the last one is used. If e.g. the first header line has a BER \( M_l \) and the last header line also has a BER \( M_l \), the last one will be used. Maximum likelihood linear fitting is used. It is assumed that both variables have normal and correlated errors. See subroutine \text{maxlik.for} in \text{LIB} for more info.

The following parameter can be selected:

- Any magnitude and agency
- Seismic moment (log)
- Stress drop (log)
- Corner frequency (log)
- Source radius (log)
- Spectral decay
- Omega zero level (log)

If any of the spectral parameters are selected, or moment magnitude is without agency, there will be an additional question about which station and component. A blank return means the average will be used.
With these parameter selections, it is possible to compare spectral parameters from any two channels, compare the average spectral parameter with the parameter from one channel etc.

Output: A plot will be shown on the screen with the observations and the least squares fit and the values are also printed out on the screen. It is now possible to click on any symbol to get up the corresponding date and time. This facility can be used to find bad magnitudes. A file mag.mag.out contains the pairs of magnitudes used.

All the events used are output in a CAT file called mag_used.out so MAG can be used to select events with two particular parameters which both are there.

3: Magnitude conversions

If a relation between two magnitude scales is known, e.g. by using option 2 above, an output file can be made with the converted magnitudes. The relation to use is specified in the mag.par file. Several different input magnitude types and agencies can be used and the relation-agency used is given in a priority list in the mag.par file, see example below. It is here shown that if a BER Mc is available, this will be the first choice. If no BER Mc then BER Mb will be the next choice etc. The new magnitude will have type X and agency NEW. Output: The output file is mag_new.out and has the same format as the input file. On the header line, the old magnitudes are removed and in the first magnitude position will be the converted magnitude (NEW) while in the second magnitude position, the magnitude selected for conversion will be given. The third magnitude position is blanked out. The conversion option can also be used to move magnitudes around by using a 1 to 1 relation as shown in mag.par example.

Summary of output files:

- **mag.amp.out**: Details each event for amplitude regression.
- **mag.coda.out**: Magnitude vs coda, see text.
- **mag.mag.out**: Pairs of magnitudes used for regression.
- **mag_new.out**: Events with converted magnitudes only.
- **mag_newa.out**: All events, both converted and non converted (due to no correct input magnitude available).
- **mag.spec.out**: Summary of normal header line, all associated magnitudes and spectral parameters.
- **mag.ml_inv.out**: From ML inversion

In DAT there is an example **mag.par** file.

An example of the **mag.par** parameter file:

```
MAGAGA is the magnitude type and agency to use for the converted magnitude.
MAGREL gives the magnitude and agency to use for conversion e.g. LBER and the parameters 2 and 3 gives the relationship:

magnitude out = magnitude in * par2 + par3
```

The magnitude conversion uses one of the MAGREL relationships, where the...
Magntitude relation: $y = 0.857x + 0.047$

**Figure 35.1:** Example of using the MAG program. Relation between NORSAR and Bergen local magnitudes.
priority is in the same order as found in the list.

SCREENOUT can be (Y)ES or (N)O, indicates if a line is printed on the screen for each event.

When doing a magnitude regression on coda or amplitude, BAD STATION indicates stations not to be used. MAG_TYP_COF is the magnitude type and agency to correlate coda readings with and the second parameter is the distance correction term used when calculating a coda magnitude relation with a fixed distance term. The same parameter is also used for amplitude regression.

The input file can be either Nordic or compact Nordic, however if coda or amplitudes are to be used, it must be NORDIC.

<table>
<thead>
<tr>
<th>KEYWORD</th>
<th>Comments</th>
<th>Par.1</th>
<th>Par.2</th>
<th>Par.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAD STATION</td>
<td>Station not used for mag relation BER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAD STATION</td>
<td>Station not used for mag relation XXX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BAD STATION</td>
<td>Station not used for mag relation BER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAGAGA</td>
<td>Magnitude type and agency, new mag XNEW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAGREL</td>
<td>Mag type, agency, a,b for new mag CBER 1.0 2.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAGREL</td>
<td>Mag type, agency, a,b for new mag BBER 1.0 2.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAGREL</td>
<td>Mag type, agency, a,b for new mag SBER 1.0 2.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAGREL</td>
<td>Mag type, agency, a,b for new mag WBER 1.0 2.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SCREENOUT</td>
<td>Header line printed each event N</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAG_TYP_COF</td>
<td>Mag. type for corr., dist coff. LNAO 0.002</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Chapter 36

ML inversion, MAG2

MAG2 is a program to invert for the local magnitude scale ML. The difference to the inversion done in MAG is that MAG2 inverts amplitudes from all events simultaneously for the scale and station corrections. The program can invert for different scale parameters depending on selected distance ranges. The reason for this is that it is known that the geometrical spreading is not the same for example between Pg and Pn. Some authors have suggested distance dependent scales, but most commonly a single scale is used for all distances for simplicity.

The general ML scale is given by

\[ ML = \log_{10} A + a \log_{10}(R) + bR + S + c \]

where we measure the displacement amplitude A in nm, R is the hypocentral distance in km, S is the station correction of the individual stations, and c is a constant added to make the scale comparable to other places at a reference distance. The station corrections add up to 0. The region dependant parameters in the scale are a, accounting for geometrical spreading, and b, accounting for attenuation. The part \((a \log_{10}(R) + bR + c)\) is commonly written as \((-\log_{10} A_0)\).

The program applies singular value decomposition using the Numerical Recipe [Press et al., 2003] routines to invert the observations for a, b and S. It then computes the parameter c based on the reference given through distance, amplitude and magnitude. This allows to calibrate scales between different regions so that they are the same at the reference distance. Commonly c is set such that 480 nm amplitude at 17 km gives ML=2 (this is equivalent to 1 mm on a Wood-Anderson seismograph giving ML=2 at 17 km [Hutton and Boore, 1987]. The original definition was 480 nm at 100 km giving a ML=3 (equivalent to 1 mm on a Wood-Anderson seismograph at 100 km distance), however it is now considred that a shorter reference distance will give a more accurate scale. The inversion can be setup to invert for the geometrical spreading term a in the scale for up to three distance ranges. However, a single attenuation term b is used.

As input the program requires a parameter file mag2.par (or other name can be given as input) in the working directory, and a standard station file e.g. STATION0.HYP. Then the user only has to enter the input file of events in Nordic format, and if needed the name of the parameter file if different from mag2.par. A sample file, mag2.par, and an input file, mag2nor.cat, with events from Norway are in given in DAT.

The parameter file has the following settings given by keywords (any order):

465
CHAPTER 36. ML INVERSION, MAG2

INVERSION TYPE (f10.1) - 1. = singular value decomposition (no other choice yet)
DISTANCES (2f10.1) - distance range in km for observations to use
MINIMUM NUMBER OF OBS/EVEN (f10.1) - only events with more or equal number of observations are used
MIN DISTANCERANGE RATIO (f10.1) - minimum range required computed as ratio of distances defined by DISTANCES
ORIENTATION (f10.1) - use of components: 0. = horizontal and vertical, 1. = horizontal only, 2. = vertical only
SYNTHETIC (f10.1) - set to 1. for synthetic test, scale defined by FIX SCALE A and FIX SCALE B; 0. for inversion of data
NOISE (f10.1) - ratio of amplitude to be added as noise to synthetic test
FIX SCALE A (3f10.1) - set the fixed parameter a in scale, possible for the three distance ranges given by SCALE DISTANCE
FIX SCALE B (3f10.1) - set the fixed b parameter in scale, possible for the three distance ranges given by SCALE DISTANCE
FIX SITE (f10.1) - set to 1. to not invert for station corrections; 0. for default inversion for station corrections
IGNORE COMP (a4) - give component not to be used
IGNORE STAT (a5) - give station not to be used
REFERENCE DISTANCE, REFERENCE AMPLITUDE, REFERENCE MAGNITUDE (all f10.1) - setup of the reference, used to calculate parameter c, give amplitude as Wood Anderson amplitude in mm
SCALE DISTANCE (2f10.1) - give up to two distances which give the transition between the possibly three distance dependent scales; blank or numbers larger than maximum distance will give only one scale RANGE A (3f10) - The parameter ‘a’ in the ML scale can be fixed to values given here. Three numbers have to be given, the first gives the start value, the second the end value and the third gives the spacing (e.g., .5,1.5,.1 means that mag2 inverts for ‘a’ values between .5 and 1.5 with .1 spacing); if this parameter is set, mag2 does run the inversion for the given (fixed) values of ‘a’.

The program produces a number of output files:

mag2.amp.dis.out - amplitude versus distance for each event
mag2.amp.obs.out - list of observed amplitudes
mag2.events_read.out - listing of events that were read in
mag2.events_used.out - events used in Nordic format
mag2.evxy.out - coordinates of events used
mag2.out - general output file, lists data used and computed scale
mag2.paths.out - event-station path coordinates
mag2.station_hyp.out - hyp station file with scale and station corrections
mag2.stat_list.out - simple output of stations used
mag2.statxy.out - coordinates of stations used

The output file mag2.out will give some details on the input data, as number of stations, events and observations. It reports the reference used to fix the scale at the reference distance. Next it gives the scale, consisting of three parts if inversion is done for all possible segments. This will be given by a1, a2, a3, while b will be assumed to be the same. First the scale is presented to include the reference distance, second it is shown without the reference distance included with the scale. Then comes a section with the stations and the respective site terms, and finally the list of events with the source term inverted for.

The output file mag2.out for the example in the DAT directory should look like this:
36.1. MAGSTAT

ML inversion output

SVD inversion

Total number of events: 69
Total number of stations: 23
Total number of observations: 600

Reference distance = 100.0000
Reference amplitude = 1.000000
Reference magnitude = 3.000000

ML = log $A + a \log(\text{dist}/\text{refdist}) + b (\text{dist} - \text{refdist}) + c + S$

- $a_1 = 0.84717 +/- 0.39844$
- $a_2 = 0.00000 +/- 0.00100$
- $a_3 = 0.00000 +/- 0.00100$
- $b = 0.00061 +/- 0.00136$
- $c = 0.31807 +/- 0.00000$

ML = log $A + a \log(\text{dist}) + b (\text{dist}) + c + S$

- $a_1 = 0.84717 +/- 0.39844$
- $a_2 = 0.00000 +/- 0.00100$
- $a_3 = 0.00000 +/- 0.00100$
- $b = 0.00061 +/- 0.00136$
- $c_1 = -1.43679$
- $c_2 = 0.25756$
- $c_3 = 0.25756$

Station # 1 STAV -0.120 +/- 0.2612 58.935 5.702
Station # 2 BLS5 -0.044 +/- 0.2204 59.423 6.456
Station # 3 ODD1 -0.085 +/- 0.2125 59.912 6.628

... Average site term: 0.00

Event # 1 2002051922484590 ML = 1.95 +/- 0.438
Event # 2 2002052614481700 ML = 1.68 +/- 0.577
Event # 3 2002063023341400 ML = 1.86 +/- 0.491

36.1 MAGSTAT

The program MAGSTAT can be used with MAG2 to produce statistics that allow to evaluate the magnitude scale. As input it takes a STATION file, as produced by MAG2, and a Nordic input file. Here, one can use any data set, or the events that were used in the MAG2 inversion.

This program is still under construction.
Chapter 37

Explosion filtering, EXFILTER

The program EXFILTER is used to identify probable explosions in a catalog of seismic events. Man-made seismic events like quarry blasts, mining explosions and other explosions show a certain distribution in time and space. Therefore the method of explosion identification here is based on normalizing the time of day distribution of seismic event occurrence as a function of area. The program works on the following principle: Areas where explosions occur are defined. If an event is located in one of these areas, with a magnitude below a given maximum magnitude, with a depth less than a given maximum depth, within a given time of day interval and within a given year interval, it is identified and marked as probable explosion. The areas are defined by polygons of any shape. For definition of the filter areas, a list of mine locations (with consideration of location accuracy), locations of explosions and locations of event clusters (they might be clearly related to mine locations, but others might indicate unknown explosion sites) can be used. The next step is to define the parameters for each area to get a normal time of day distribution. They can be determined following the steps:

1) - get the time of day distribution of events (program CATSTAT)
   - select a time window of probable explosions
   - select events within time window of probable explosions

2) - get the distribution of magnitudes of events within time window of probable explosions (program BVALUE)
   - select the maximum magnitude

3) - test parameters defined with program EXFILTER for the defined area and adjust the parameters if the time of day distribution is not normal.

For more details, see [Ottemöller 1995].

The program uses a parameter file, EXFILTER.PAR which MUST be located in the DAT directory.

An example of the parameter file EXFILTER.PAR

--------------Parameter file for program EXFILTER-------------------
------------------------------------------------------------------
This file must follow the following format rules: __________
1. Any number of comment lines
2. Any line with first character # defines the parameters of that area
3. Any line with first character * defines the parameters of exceptions within an already defined area. I.e. you can define an area around a volcano, and make an exception for that very small area.
4. In the first line of each parameter set the parameters are:
   Maxdepth: Events with depth above this value are not explosions
   Maxmag: Events with magnitude above this value are not explosions
   Lhour: Lower limit of time interval for explosion time
   Hhour: Upper limit of time interval for explosion time
   Btime: Lower limit of yearly interval
   Etime: Upper limit of yearly interval
   N: Number of latitude longitude pairs in polygon
5. From the second line the lat long pairs are given

<table>
<thead>
<tr>
<th>Area</th>
<th>Maxdepth</th>
<th>Maxmag</th>
<th>Lhour</th>
<th>Hhour</th>
<th>Btime</th>
<th>Etime</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td># area 1</td>
<td>100.0</td>
<td>3.6</td>
<td>11</td>
<td>15</td>
<td>197001</td>
<td>200012</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>64.0</td>
<td>23.8</td>
<td>66.70</td>
<td>23.80</td>
<td>68.80</td>
<td>20.20</td>
<td>69.80</td>
</tr>
<tr>
<td></td>
<td>70.50</td>
<td>20.0</td>
<td>71.20</td>
<td>26.00</td>
<td>70.80</td>
<td>29.50</td>
<td>70.00</td>
</tr>
<tr>
<td></td>
<td>68.50</td>
<td>28.50</td>
<td>67.00</td>
<td>29.50</td>
<td>64.30</td>
<td>29.50</td>
<td>64.00</td>
</tr>
<tr>
<td># area 2</td>
<td>100.0</td>
<td>3.8</td>
<td>15</td>
<td>20</td>
<td>197001</td>
<td>200012</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>67.90</td>
<td>17.80</td>
<td>68.80</td>
<td>20.20</td>
<td>66.70</td>
<td>23.80</td>
<td>66.50</td>
</tr>
<tr>
<td># area 3</td>
<td>100.0</td>
<td>3.5</td>
<td>0</td>
<td>1</td>
<td>197001</td>
<td>200012</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>67.90</td>
<td>17.80</td>
<td>68.80</td>
<td>20.20</td>
<td>66.70</td>
<td>23.80</td>
<td>66.50</td>
</tr>
<tr>
<td># area 4</td>
<td>100.0</td>
<td>4.0</td>
<td>2</td>
<td>17</td>
<td>197001</td>
<td>200012</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>68.50</td>
<td>28.50</td>
<td>68.50</td>
<td>36.00</td>
<td>67.00</td>
<td>36.00</td>
<td>67.00</td>
</tr>
</tbody>
</table>

The EXFILTER program searches for probable explosions using a catalog-file as input and marks events that might be explosions with 'P' as Event ID in the output file exfilter.out. Example of program run

<exfilter>

NUMBER OF AREAS: 55

FILENAME... ?
june.cat

************************************
Number of probable explosions found: 90
Output written in file: exfilter.out

************************************
Figure 37.1: Figures that show how the exfilter works for events in Scandinavia:
The top figure shows the filter areas used for Scandinavia. The bottom right figure shows
the time of day distribution for a 10 year Scandinavian catalog before filtering (made
with CATSTAT) and the figure bottom left shows the distribution after filtering.
Chapter 38

Inversion of travel time data and joint hypocenter determination

38.1 VELEST

Introduction

The program VELEST is used to solve the coupled hypocenter velocity model problem for local earthquakes. It performs a simultaneous inversion for hypocenters and velocity model. The inversion is limited to first arriving phases. A detailed program description is given in the ‘VELEST USER’S GUIDE’ [Kissling 1995]. A recipe for preparing data and use of the inversion routine is presented in ‘Initial reference models in local earthquake tomography’ [Kissling et al. 1994]. The two documents are available in one Postscript file in the INF directory, the filename is ‘velest.ps’. The derived model can be used as an improved model for earthquake location or as a starting model for 3-D inversion. For a fixed velocity model and constant station corrections, VELEST in simultaneous mode performs the Joint-Hypocenter-Determination (JHD).


The original version of VELEST by Kissling is included in the Sun and Linux versions. A version modified to compile VELEST under Windows has been provided by Freddy Aldersons (e-mail: faldersons@earthlink.net). This Windows package is included in the file velest_pc_3.3.zip, which is located in the SUP directory. The files have to be extracted to the PRO directory.

The implementation of VELEST to SEISAN is given by the program VELMENU. VELMENU provides:

- automatic format conversion to VELEST
- generation of parameter files using the SEISAN system
- execution of VELEST
- conversion back to SEISAN format

After preparing a dataset of local earthquake data, VELMENU can be used to work with the VELEST
inversion routine. The first time VELMENU is used, all input files for the inversion with default parameters can be generated. These parameter files then can be changed interactively and the inversion started with VELMENU.

**Running VELMENU**

The program is started with ‘velmenu’. After entering the filename of the earthquake data the menu of VELMENU appears.

Example of program run:

```
velmenu

File name of earthquake data in Nordic Format:
select.out

VELEST MENU

1. Create VELEST command file (velest.cmn)
2. Edit/change VELEST command file (velest.cmn)
3. Create station select file (selstat.lis)
4. Edit/change station select file (selstat.lis)
5. Create model file
6. Edit/change model file
A. RUN VELEST
B. Edit inversion output file
C. Convert VELEST output to Nordic format and make diff-file
Q. End

Choice?
```

The complete inversion-process of earthquake data in SEISAN format, including all conversions and preparation of parameter files, can be done with VELMENU. The steps are as follows:

1: Create VELEST command file (velest.cmn)
   The user is asked for inversion or JHD and the appropriate parameters are set. The file velest.cmn is the central VELEST parameter file. To create it, the file of earthquake data is read to determine the parameters that depend on the data. These are the number of events and the center of Cartesian coordinate system, which is simply determined as the average of latitude and longitude of epicenter locations. The remaining parameters are set to default values.

3: Create station select file (selstat.lis)
   For the inversion, VELEST will use phases from stations with an epicentral distance below a maximum distance only. In addition in VELMENU a selection of stations has to be used, only phases from stations given in the file selstat.lis will be used for inversion. When generating the file, the maximum distance between station and hypocenter (parameter ‘dmax’) is read from velest.cmn and the input data are scanned to get a list of stations, which are within the limit to any epicenter. Editing the file, stations can be added or removed. If all stations should be used for inversion, the parameter ‘dmax’ in the file velest.cmn has to be increased.

Example of selstat.lis:
38.1. VELEST

# # STATION SELECT FILE FOR PROGRAM VELEST
# # STATIONS WILL BE USED IN THE VELEST
# INVERSION PROGRAM
# # COMMENT LINES START WITH
#
#
KONO
BER
NRAO
...

NOTE: The order of the stations is as given by the input data file. VELEST uses the last station as reference station, so you may want to change the order.

5: Create model file
The input model file ‘model.inp’ is created using the model as given in the ‘STATION0.HYP’ file. The ‘STATION0.HYP’ file, if available, will be read from the local directory, otherwise from the DAT directory. This might be a reasonable starting model, but of course the model file has to be changed.

A. RUN VELEST

Once the parameter files are created the inversion program can be started. The inversion study requires interactive changing of parameters, which is supported by VELMENU. All input parameter files can be changed from VELMENU. NOTE: ‘... please accept the warning: To calculate a Minimum 1-D model a single or even a few VELEST runs are useless, as they normally do not provide any information on the model space!’ [Kissling, 1995]. The conversions and the inversion programs are started as one process.

Before the inversion routine is started the station locations will be converted from the STATION0.HYP file and the earthquake data in Nordic format will be converted to CNV (hypocenters and travel times) format. NOTE: VELEST does not support 5 character station codes, therefore in the conversion to VELEST, only the first 4 characters are used if the station code has 5 characters. In the conversion of the earthquake data only phase readings from stations included in the station selection file will be used. Arrivals with a time residual, given in the Nordic input file, above five seconds are omitted. Only the first arriving phase of P and S respectively are used. The hypocenter location given by the inversion will be determined by first arrivals only. The original data might include more phases like Pg, Sg or Lg. Therefore, to get a comparison of hypocenter locations between the HYP location program and VELEST, a Nordic file including the same data as the CNV file is created and the HYP program run on this file before VELEST is started. The HYP program can be skipped by pressing ‘CTRL+C’, while it is running.

The results of the inversion will be given in a text file that can be viewed within VELMENU. VELMENU provides an option to convert the VELEST output file with final hypocenter locations in CNV format back to Nordic format and to write a file that shows differences (velout.dif) in location and time between the two location routines, HYP and VELEST, based on the same input data.

Example of velout.dif:

differences: inversion output - SEISAN input
first line input, second line output, third line difference
CHAPTER 38. INVERSION OF TRAVEL TIME DATA AND JOINT HYPOCENTER DETERMINATION

1984 8 5 0235 22.7 L 59.449 4.968 0.0 BER 9 1.3 1.9CBER 3.7BBER 1
1984 8 5 235 22.1 L 59.458 5.140 0.1 BER 9 1

time: -0.6 latitude: 0.009 longitude: 0.172 depth: 0.1

1984 8 6 0406 26.6 L 59.538 5.678 1.9 BER 9 1.8 1.7CBER 3.6BBER 1
1984 8 6 4 6 26.6 L 59.505 5.655 6.2 BER 9 1

time: 0.0 latitude: -0.033 longitude: -0.023 depth: 4.3

Files will be overwritten, when VELMENU is started again. To work with different datasets or parameter files it is recommended to work on different directories or to change the filenames, but note that the default filenames (see below) will be used in VELMENU.

**Problems:** VELEST skips events without phase readings and therefore the number of events read by VELEST will be different from the number given in the `velest.cmn` file. If this is the case VELEST stops with the message STOP: ...end...(VELEST was running with the SINGLE-EVENT-OPTION). Events without phase readings will not be listed in the `invers.out` file, and should be deleted from the input file.

**Joint-Hypocenter-Determination (JHD)**

VELEST for fixed velocities and station corrections can be used as a JHD routine. For JHD, VELMENU is used in the same way as described above for inversion. The only difference is that when generating the `velest.cmn` you have to choose JHD. The appropriate file for JHD is then generated. Some parameters in the `velest.cmn` file are different, compared to the inversion. These are `dmax`, `nsinv` and `invertratio`, see ‘VELEST USER’S GUIDE’ for details. The output of final hypocenter locations as described above can be converted to Nordic format, but note that the JHD will be based on first arriving phases only.

Example of JHD:

File name of earthquake data in Nordic Format : select.out

VELEST MENU

1. Create VELEST command file (velest.cmn)

Choice ?

1

Inversion or JHD (I / J) ?

j

Origin of cartesian coordinates : 59.394 -5.707

number of events: 61

**Potential problem:** We have seen cases where in JHD mode the depth parameter in the inversion is sensitive to invertratio, which when set to 1. in JHD means that VELEST inverts for station correction in every iteration. VELEST in this case worked better with an invertratio of larger than 1. See VELEST manual for details.
List of files generated by VELMENU / VELEST

data.cnv  earthquake data in CNV format, VELEST input, generated by VELMENU
data.nor  earthquake data in Nordic format, HYP input, generated by VELMENU
fin_hyp.cnv final hypocenter locations in CNV format, VELEST output
hyp.sum.out earthquake data in Nordic format, HYP output
hypsum.out HYP output file
input.mod  input model, VELEST input, generated by VELMENU
invers.out documentation of inversion, VELEST output
nor1.date earthquake data in Nordic format, VELMENU input
print.out  HYP output file
selstat.lis selection of stations, generated by VELMENU
sta.cor.out station corrections, VELEST output
station.sta station locations, VELEST input, generated by VELMENU
velout.dif difference file between HYP and VELEST location routine, VELMENU output
velout.nor final hypocentre locations, same as fin_hyp.cnv, in Nordic format,
VELMENU output

```
velest.cmn  VELEST control file, VELEST input, generated by VELMENU
```

### 38.2  NOR2DD

By *Brian Baptie*, BGS

This program produces input for the Double Difference Program hypoDD \[Waldhauser, 2001; Waldhauser and Ellsworth, 2002\] from Nordic and STATION0.HYP files. The Nordic file has to be given as argument when running the program, example:

```
nor2dd select.out
```

The files created are:

- **phase.dat**: phase input data
- **station.dat**: station coordinates


### 38.3  hypoDD

This program Waldhauser and Ellsworth (2000) is meant to improve relative locations in clusters of events. Quoting the manual (hypod.pdf in INF):

"*The DD technique (described in detail in Waldhauser and Ellsworth, 2000) takes advantage of the fact that if the hypocentral separation between two earthquakes is small compared to the event-station distance and the scale length of velocity heterogeneity, then the ray paths between the source region and a common station are similar along almost the entire ray path. In this case, the difference in travel times for two events observed at one station can be attributed to the spatial offset between the events with high accuracy.*"
Thus arrival time differences between stations is used. Data from catalogs or cross correlation can be used, here we will only illustrate the use of catalog data. For details, see the manual. hypoDD is installed in SEISAN and can also be obtained from http://geopubs.wr.usgs.gov/open-file/of01-113/. For simplicity, the program is called hypodd and not hypoDD. To run the program, the steps are:

1: Select a cluster of events.
2: Create an input file to hypodd program ph2dt with SEISAN program NOR2DD. NOR2DD takes a CAT file (like select.out) and makes a file phase.dat (default input file to PH2DT).
3: Created difference times with PH2DT (supplied with hypodd, included in SEISAN).
4: Run HYPODD.
5: Plot the new and original locations with EPIMAP, output is provided in Nordic format. Command is epimap hypodd.map

NOR2DD
The program takes an input CAT file and creates an output file phase.dat (default name) with the arrival times and hypocenter information:

```
#  2015  4  13  15  32  40.6  66.6120  13.3800  15.1  0.7  5.2  9.4  0.6  1
KONS  3.13  1.0 P
KONS  5.94  1.0 S
STOK  7.02  0.5 P
STOK  11.13  1.0 S
MOR8  19.50  1.0 S
FAUS  18.71  0.5 P
FAUS  33.65  1.0 S
STEI  42.45  1.0 S
```

The phases P and S are assumed to be first arrivals, whether Pg or Pn. There are 2 options for selecting phases, either use all phases P and S irrespective of given names so e.g. Pn becomes P or only use phases labeled P and S in input file. It is recommend to use all phases since phases labeled Pn and Pg most likely are first arrivals. Any double entry of a phase will be eliminated by giving the latest arrival phase zero weight. In this way a Pg following a Pn for the same station will be weighted out.

Optionally, phases from given stations can be given a different weight than given in the S-file. This can be used to remove station by giving weight 0 or lowering the weight. The weigh -9 can also be given to indicate that this station must be used in all cases (see hypodd manual).

The program also creates the station file for HYPODD using the STATION0.HYP (in local or DAT directory) with the default name station.dat:

```
KONS  66.4990  13.1177
STOK  66.3330  13.0177
MOR8  66.2852  14.7317
FAUS  67.3820  15.2877
STEI  67.9300  15.2420
```

Once the location is made with HYPODD, the output locations can be converted to Nordic format with NOR2DD. It is assumed that the default file hypoDD.loc is used for the original locations and
38.3. HYPODD

hypoDD.reloc is used for the relocations. The output file with the relocated events is called hypodd.out and the corresponding original locations is hypodd.org. Hypodd.org will usually contain fewer events than the original input in phase.dat since events are eliminated during the relocation process, see later.

Examples:

nor2dd
   Convert from Nordic to hypodd (1=enter) or hypodd to Nordic (2)

   Give input Nordic file
swarm.out
   Use any P and S-phase (enter) or only P and S (o)

   Stations and weights 0-1 or -9 to force use, enter to terminate
NBB13 0

   The input file had 388 events
   Output phases in file phase.dat
   Output of station list in station.dat

nor2dd
   Convert from Nordic to hypodd (1=enter) or hypodd to Nordic (2)
2
   Give input hypodd file, hypoDD.reloc is default (enter)

   Number of events in original file 39
   Number of events in relocated file 24
   Output file of relocated events is hypodd.out
   Output file of corresponding original events is hypodd.org

PH2DT

PH2DT find the stations pairs to be used. It has a default parameter file ph2dt.inp. The most important parameters are event separation and maximum distances. Since these parameters also can be set on the HYPODD parameter file, hypodd.inp, the simplest is to leave ph2dt with default parameters. That includes the default phase data file phase.dat and station.dat made with NOR2DD:

* ph2dt.inp - input control file for program ph2dt
* Input station file:
  station.dat
* Input phase file:
  phase.dat
  *MINWGHT: min. pick weight allowed [0]
  *MAXDIST: max. distance in km between event pair and stations [200]
  *MAXSEP: max. hypocentral separation in km [10]
  *MAXNGH: max. number of neighbors per event [10]
  *MINLNK: min. number of links required to define a neighbor [8]
  *MINOBS: min. number of links per pair saved [8]
*MAXOBS: max. number of links per pair saved [20]
*MINWGT MAXDIST MAXSEP MAXNGH MINLNK MINOBS MAXOBS
0 200 10 10 8 8 20

To run the program, write
ph2dt ph2dt.inp

HYPODD

The program has a large number of parameters and it can be a bit confusing to adjust for best performance, see manual for details. The most important seem to be the event separation WDCT (see example parameter file below and example runs), weight for P and S-data WTCTP and WTCTS, respectively and cut-off threshold for outliers WRCT. Inversion is normally with conjugate gradients but for a small data set (less than 40-50 events), least squares solution can be used. The parameter file must also be set up for the model used.

Running HYPODD
Hypodd hypodd.inp

For explanation of output, see hypodd manual. The program will, in the SEISAN version, also execute the NOR2DD program to generate, in Nordic format, the files hypodd.out and hypodd.org as describe under NOR2DD. The result can then be plotted immediately.

38.3.1 hypoDD example

We will illustrate the use of HYPODD with an example data set from the Norwegian National Seismic network. The data consist of a swarm of event occurring mostly in 2016. All data and parameter files are available in the test data set under WOR/hypodd. The original data consists of 388 events located in an area of about 10km in diameter, see Figure 38.1.

Despite that two stations (NBB13 and NBB15) are within 5-10 km of the swarm, the epicentre pattern is diffuse. Likewise the depths, where many are at the unlikely depth of zero km. Some testing was done to obtain a best solution in terms of number of events remaining and RMS of travel time residuals, see Figure 38.2. The corresponding parameter file is listed below.

It is seen that the epicenters are now in 3 groups with the southernmost being the most clear. The depths are now much more clustered with the southernmost group the most concentrated. Nearly all zero depth events have been removed or relocated.

Now some of the parameters were then varied, one by one, to see the effect on the results.

Separation distance

One of the more sensitive parameters is the event separation distance (WDCT). For the reference solution, 2 km was used for the 2. set of parameters and 5 for the first set. However with such a small cluster, one could think that 1km would be more reasonable so this was tested for the 2. set of parameters, see Figure 38.1 for the result.

The clustering has not changed much but much less results are obtained. A test was also made with 4km (276 events), however clustering was much less clear. So it seems that the original 2km separation is ok. A test was also made to increase WDCT to 10km in first data set, but results were worse.

Number of iterations

The number of iterations should be enough that the RMS does not decrease anymore.
Figure 38.1: Dataset for testing HYPODD. To the left is seen the epicentres and to the right the depth profile taken from SW to NE as outlined on the left figure. There are 388 events.
Figure 38.2: Results of the hypodd inversion with best parameters. The number of events have been reduced to 248. The red color indicates the original solutions and the green the hypodd solutions.

Figure 38.3: Testing with event separation distance of 1 km. There are 158 events.
With fewer iterations, there were more events, but the clustering is less clear. More iterations were also tested, but with no significant difference. In this data set 8+8 iterations seems ok.

Maximum distance

In the standard parameter set the maximum distance it is set to 200km. Considering that far stations might have less clear arrivals than near station, using in shorter distance might improve the results. Testing with 80k max distance is seen in Figure 38.5. Only 4 iterations were needed in the 2 parameter set.

Clearly the clustering is less clear than when using standard parameters so larger distances should be used for this data set.

Importance of nearest station

Station NBB13 is very close to the southern cluster so a test was made weighting out the station (with NOR2DD).

The clustering is now less clear so one near stations can change results significantly. But the results are still better with HYPODD than using standard locations. NBB13 has readings for 292 events out of the original 388. However of the 248 events with the standard parameters, 225 events had station NBB13 so the station is clearly important for the best solutions.

Other tests: Using only P resulted in fewer events (232) with more scatter, maybe not surprising. The outlier parameter WRCT did not affect the results much provided they were keeps within ‘reasonable’ limits as in our example.

These tests are mostly valid for the current data set, other data sets might require other parameters. But the test should give an indication of the importance of the different parameters. In many cases there is a
Figure 38.5: Testing with maximum distance of 80 km instead of 200 km. There are 231 events.

Figure 38.6: Results without using station NBB13. There were 212 events.
trade-off between number of events which can be relocated and the demand for stricter parameters. For a better understanding, read the hypodd manual and paper.

Hypodd.inp file for the best solution.

* RELOC.INP:
*---- input file selection
* cross correlation diff times:
* catalog P diff times:
dt.ct
*
* event file:
event.sel
*
* station file:
station.dat
*
*---- output file selection
* original locations:
hypoDD.loc
* relocations:
hypoDD.reloc
* station information:
hypoDD.sta
* residual information:
hypoDD.res
* source parameter information:
hypoDD.src
*
*---- data type selection:
* IDAT: 0 = synthetics; 1= cross corr; 2= catalog; 3= cross & cat
* IPHA: 1= P; 2= S; 3= P&S
* DIST:max dist [km] between cluster centroid and station
* IDAT  IPHA  DIST
  2  3  200
*
*---- event clustering:
* OBSCC: min # of obs/pair for crosstime data (0= no clustering)
* OBSCT: min # of obs/pair for network data (0= no clustering)
* OBSCC  OBSCT
  0   0
*
*---- solution control:
* ISTART: 1 = from single source; 2 = from network sources
* ISOLV: 1 = SVD, 2=lsqr
* NSET: number of sets of iteration with specifications following
* ISTART  ISOLV  NSET
  2   2   2
*
*---- data weighting and re-weighting:
* NITER: last iteration to used the following weights
* WTCCP, WTCCS: weight cross P, S
* WTCTP, WTCTS: weight catalog P, S
* WRCC, WRCT: residual threshold in sec for cross, catalog data
* WDCC, WDCT: max dist [km] between cross, catalog linked pairs
* DAMP: damping (for lsqr only)

* --- CROSS DATA ---- ----CATALOG DATA ----*
* NITER WTCCP WTCCS WRCC WDCC WTCTP WTCTS WRCT WDCT DAMP
  8 1 -9 -9 -9 1.00 0.5 5 5 20
  8 1 -9 -9 -9 1.00 0.2 2 2 20

*--- 1D model:
* NLAY: number of model layers
* RATIO: vp/vs ratio
* TOP: depths of top of layer (km)
* VEL: layer velocities (km/s)
* NLAY RATIO
  6 1.74
* TOP
  0.0 12.0 23.0 31.0 50.0 80.0
* VEL
  6.2 6.6 7.1 8.05 8.25 8.5

*--- event selection:
* CID: cluster to be relocated (0 = all)
* ID: cuspids of event to be relocated (8 per line)
* CID
  1
* ID

38.4 NOR2SIMULPS, NOR2SIMULR

By Felix Halpaap, UIB

These programs produce input for the programs simulps and simulr from Nordic and STATION0.HYP files. The Nordic file has to be given as argument when running the program, example:
nor2simulps select.out nor2simulr select.out

The files created are:
EQKS: phase input data
STNS: station coordinates

38.5 NOR2JHD_PUJOL

By Brian Baptie, BGS
Figure 38.7: NO CAPTION FOR THIS FIGURE
This program produces input for Pujol’s JHD program [Pujol 2003] from Nordic and STATION0.HYP files. The Nordic file has to be given as argument when running the program, example:

```
nor2jhd_pujol select.out
```

The files created are: `syn.times`: phase input data  
`syn.vel`: velocity model  
`stalist.elev`: station list

The inversion tool is available from:  
ftp://beagle.ceri.memphis.edu/pub/pujol/JHD  
Chapter 39

Analysis of volcanic earthquakes

SEISAN is often used for volcanic monitoring. Many of the standard tools used in SEISAN can also be used for volcanic earthquakes, like epicenter location and magnitude. However, more special tool are also needed and below there is a description how this is done at the British Geological Survey (BGS).

Another description, made by Andrew Lockhart at the USGS, is given in a separate manual (seisan_volcano.pdf in INF or at [http://seis.geus.net/software/seisan/seisan_volcano.pdf](http://seis.geus.net/software/seisan/seisan_volcano.pdf)). This manual is a mini SEISAN manual detailing the steps under Windows.

Volcano monitoring at BGS

By Brian Baptie

Background

An important part of volcanic seismology and the seismic monitoring of active volcanoes is the correct recognition of the different types of seismic event generated by the volcanic activity. The principal event types include, volcano-tectonic events, caused by shear or tensile failure of rocks; long period events, generated by a volumetric source in a liquid; hybrid events; and volcanic tremor.

To be of value for volcanic monitoring, any database of seismic events should include the type or sub-class of individual events. This should allow users to then extract phase and location information over a selected time period for individual event types and calculate hourly and daily rates of event. Simple histogram plots showing the distribution of subclasses over time can be generated with the program VOLCSTAT (Unix only).

Initialization

The user should create a text file in the DAT directory called VOLCANO.DEF (an example is already in the directory). The format of this file will be one line of text (80A) followed by successive lines with the format "i2,1x,6A,1X,40a" for number, sub-class code and description. An example of the file is shown below. Comments are preceded with '!'.

Current volcano sub-classes: ! Comment line 80 characters
1 vt volcano-tectonic ! Individual sub-class line
2 hybrid hybrid
3 lp long-period
4 tremor volcanic tremor
5 rf rockfall
CHAPTER 39. ANALYSIS OF VOLCANIC EARTHQUAKES

![Image of a volcano and earthquake]

6 un unknown
7 QUIT ! The last line should contain
         this entry

Registering volcanic sub-classes

Registration should be carried out as normal in MULPLT. From multi-trace mode enter 'p' to create a
new s-file for the event in the database. Answering 'LV' to the prompt for event type marks the event as
a local volcanic in the headers. If the VOLCANO.DEF file has been set up correctly in the DAT directory,
the information on the different sub-classes will be printed to the terminal. Choosing an appropriate
number selects the volcanic sub-class. The sub-class code is then entered in the s-file.

Modification of the s-file to incorporate volcanic sub-classes

The volcanic sub-class information is stored in a type 3 line within the s-file, e.g.

VOLC MAIN tremor 3

| Columns 2:10 'VOLC MAIN' | Header identifier |
| Columns 12:17 a6         | Sub-class flag    |
| Column 80 '3'            | line type identifier |

This allows the use of a maximum 6-character sub-class identifier, e.g. 'hybrid', which can then be
searched for and selected.

VOLCSTAT: Creating histogram plots

The program reads S-files directly from the database, and creates input files as well as a GMT script to
produce histogram plots of the distribution of subclasses over time. The user needs to enter database
name, start and end time, and the subclasses that are to be plotted. An example of a plot is shown in
Figure 39.1. The program supports 1-char subclass names only.

The following output files are created:

- volcstat.batch - c-shell script to generate Postscript output using GMT
- volcstat_counts.<type>.ps - Postscript output file
- volcstat_counts.<type>.out - for each event, the Julian date is written out, one file per subclass
- volcstat_daily.<type>.out - number of events per day, files written for each subclass
- volcstat_counts_total.out - total event counts for each subclass

RSAM

1-minute RSAM data can be created with WAVETOOL.

Future Extensions:

It is intended that additional parameters can be included in the above structure to included routine
measurements of the volcanic earthquakes. For example, signal duration, peak amplitude and mean
frequency can be calculated for individual stations and included on additional type 3 lines with a volcanic
identifier. Parameters on each channel can then be averaged an inserted on the volcanic header line.

The proposed format for these lines is as follows

<table>
<thead>
<tr>
<th>column</th>
<th>format</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2:5</td>
<td>a4</td>
<td>'VOLC' Volcanic identifier</td>
</tr>
</tbody>
</table>
Figure 39.1: Bar diagrams showing distribution of events of different subclasses over time.
7:10  a4  station
12:15  a4  component
19:20  a2  'PA' field identifier
22:29  g8.3  peak amplitude
31:32  a2  'DU' field identifier
34:41  g8.3  signal duration
43:44  a2  'MF' field identifier
46:53  g8.3  mean frequency
55:56  a2  'SB' field identifier
58:65  g8.3  signal bandwidth
66:79  blank
80  a1  '3' line type identifier.

For example

VOLC MAIN tremor
VOLC KTK1 S Z PA .152E 06 DU 1.325 MF 2.472 3
VOLC KTK1 S N PA .167E 06 DU 1.997 MF 2.067 3
VOLC KTK1 S E PA .141E 06 DU 1.543 MF 1.998 3

This method of inclusion of volcanic parameters should allow for future flexibility such as incorporation of an additional parameter fields in columns 66 to 79. Also the use of type 3 lines means that existing software, such as the update program, are unaffected by these lines.
Chapter 40

FK Analysis

The FK routines were provided by Tormod Kvaerna from NORSAR and implemented into SEISAN by Andrius Pacesa.

Some basics

The FK-analysis, more strictly slowness analysis, is a standard tool in seismic array processing. It is used to find the apparent velocity and back azimuth of an incoming wavefront. Apparent velocity can be used to identify the type of wave (P, S, Lg and etc.) and the approximate distance to the source can be determined for teleseismic events. Utilizing azimuth and distance to the source, one can define the approximate location of the signal source.

A description of frequency-wavenumber analysis - ï-k analysisï may be found in Capon (1969). This method has been further developed to include wide-band analysis and maximum-likelihood estimation techniques - see Kvaerna and Doornbos [1986].

The principle of slowness analysis is beamforming in the frequency domain for a number of different slowness values and calculating the power for each beam. The beam power will be a maximum in case the slowness of the beam coincides with the slowness of the wavefront crossing an array. So the beam having the maximum power will indicate the slowness of the incoming signal.

Running the program

The FK program can be started directly with command fk or from MULPLT. The program expects that the file waveform.out with the seismic traces as input data, is available in the current directory. If the program is invoked directly, this file has to be created before using mulpft, selecting a window and creating the ‘waveform.out’ file.

In general it is more useful, to start the FK program from MULPLT since the input file needs to be created by mulpft. The result of the fk analysis can be saved to the S-file.

The steps are:

- start MULPLT
- select channels and a time window
- use option fk to start FK program (this option creates file ‘waveform.out’ and starts FK program), accept maximum or pick value with mouse

493
The options in FK are:

R-Redo: Repeat fk analysis with different parameters
M-Mouse: 'm' or mouse click to pick values different from maximum
S-Save and quit: save picked value to file and quit
Q-Quit: quit

- use option 'save and quit' to save your result, so that it can be used by MULPLT
- back in MULPLT: pick phase on the first trace used, to store back azimuth and apparent velocity in the S-file
- in case of teleseismic events, the apparent velocity can be used for location, the fk analysis has to be done on the P phase

Note: The FK program only works by default with station file 'STATION0.HYP'. If coordinates are in e.g. STATIONt.HYP, the user will be asked to specify another station file letter, in this case ‘t’.

Example

Input:

Enter lower frequency, <ENTER> for default value 1 Hz
Enter higher frequency, <ENTER> for default value 5 Hz
Enter max slowness, <ENTER> for default value 0.4 s/km
Enter No of grid points, <ENTER> for default value 51

Do you want to plot level numbers, N/Y, <ENTER>=Y

It will take some time ...

APPARENT VELOCITY 10.26 AZIMUTH 140
NORM. POWER MAX 0.99 POWER MAX IN dB 72.44
MAX X SLOWNESS 0.06 MAX Y SLOWNESS -0.07
LOW FREQUENCY 0.94 HIGH FREQUENCY 5.00
QUALITY (1=best, 4=worst) 1
Plot file is fk.eps

Example of output file fk.out:

DATE, TIME AND WINDOW LENGTH 99 256 9 13 12 0 33.843 7.62
APPARENT VELOCITY 10.26 AZIMUTH 140
NORM. POWER MAX 0.99 POWER MAX IN dB 72.44
MAX X SLOWNESS 0.06 MAX Y SLOWNESS -0.07
LOW FREQUENCY 0.94 HIGH FREQUENCY 5.00
QUALITY (1=best, 4=worst) 1
VALUES TO SAVE
APPARENT VELOCITY 10.26 AZIMUTH 140
Station Long Lat Elev Xcoord Ycoord Zcoord
<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NRA0</td>
<td>S</td>
<td>Z</td>
<td>11.54150</td>
<td>60.73533</td>
<td>300</td>
<td>0</td>
</tr>
<tr>
<td>NRA1</td>
<td>S</td>
<td>Z</td>
<td>11.54233</td>
<td>60.73650</td>
<td>291</td>
<td>45</td>
</tr>
<tr>
<td>NRA2</td>
<td>S</td>
<td>Z</td>
<td>11.54333</td>
<td>60.73433</td>
<td>311</td>
<td>99</td>
</tr>
<tr>
<td>NRD4</td>
<td>S</td>
<td>Z</td>
<td>11.56333</td>
<td>60.72717</td>
<td>379</td>
<td>1186</td>
</tr>
<tr>
<td>NRD5</td>
<td>S</td>
<td>Z</td>
<td>11.54750</td>
<td>60.72217</td>
<td>348</td>
<td>326</td>
</tr>
<tr>
<td>NRD6</td>
<td>S</td>
<td>Z</td>
<td>11.52883</td>
<td>60.72334</td>
<td>352</td>
<td>-688</td>
</tr>
<tr>
<td>NRD7</td>
<td>S</td>
<td>Z</td>
<td>11.51617</td>
<td>60.73017</td>
<td>337</td>
<td>-1377</td>
</tr>
<tr>
<td>NRD8</td>
<td>S</td>
<td>Z</td>
<td>11.51667</td>
<td>60.73900</td>
<td>301</td>
<td>-1349</td>
</tr>
</tbody>
</table>

XSLow YSLow Power

0.40 0.40 0.05
0.38 0.40 0.05
0.37 0.40 0.05
0.35 0.40 0.05

...
Figure 40.1: The FK program can be started from MULPLT. The traces shown were selected and used as input to the FK program. The result of the FK analysis is shown in Figure 40.2. The event shown here is part of the test data set.
Figure 40.2: Output from the FK program. Contours and values are the normalized maximum power.
Chapter 41

Instrument response

In SEISAN the instrument response can be stored as pairs of frequency, amplitude and phase or as poles and zeros. The formats that can be used include GSE2, SEISAN, SAC and SEED. The SEISAN, SAC and GSE response formats are described in Appendix C. For a detailed description of the GSE format, the reader is referred to [1997]. The program RESP creates response files in SEISAN and GSE format. SEED format response files can be extracted from a SEED volume.

The response information gives the gain of a channel in counts/m and to get the correct ground displacement, the count values must be divided by the response values. In the current SEISAN system, only the programs MULPLT, WAVETOOL and SPEC use the response information when doing spectral analysis, generating Wood Anderson or ground motion traces. The programs will look first in the CAL (or alternative) directory for a valid response file and if not found there use the header information in the waveform file. A message will be given if the file header information is used.

If waveform files are generated on the SEISAN system from raw field station files or other input files without response information, the conversion programs (e.g. QNXSEI from a SEISLOG QNX system) will look in the CAL (or alternative) directory to find the response information to include with the Seisan waveform file. The response will be only put into the SEISAN waveform file, if the response is stored in SEISAN format. The response files are generated with RESP, see below and Appendix C.

The instrument response can be defined for each channel of digital data in either SEISAN, GSE, SAC or SEED (ascii version) response format. There are three places in the system where it can be stored. Often the instrument response is part of each channel header in the digital waveform file in SEISAN waveform format (see the Appendix for format description). However, the instrument response is often not available at the moment the data arrives, or it is later discovered that the response given in the waveform file is wrong. There is therefore by default a directory CAL that contains one response file for each channel and for each date from which it is valid. Since the filenames contain the date from which a change in the response was made and the channel code and component code, a directory listing of CAL will give the history in chronological order of the response of a given channel. This is the most common way to use the response information in SEISAN. Response information can also be kept in any other directory specified with the environmental variable LOCAL_CAL. The variable must be set with the full path to the directory e.g. /home/seismo/WOR/test/ or on PC, C:\seismo\new\cal\. On Sun it can be set in the SEISAN.csh file and under Windows by using the setting of environmental variables Control panel/system/advances/environmental variables. The variable can also be set from the keyboard (Linux/Sun: 'setenv LOCAL_CAL=directory', PC, 'set LOCAL_CAL=directory'. This is a useful option when testing response files. The file name of the SEISAN, GSE and SAC response files defines the start
time of which they are valid. If the response of a station is changed, a new response file must be made
with a new time stamp.

The SEED response files are named e.g. RESP.IU.TRIS.10.BHZ (see rdseed below), where the network,
station, location and channel codes are part of the filename. When SEISAN looks for a SEED response
files it will search for a file with a name matching the network, station, location and channel code of the
data in memory.

The file name of the SEISAN, GSE and SAC response files defines the start time of which they are valid.
If the response of a station is changed, a new response file must be made with a new time stamp.
The SEED response files are named e.g. RESP.IU.TRIS.10.BHZ (see rdseed below), where the network,
station, location and channel codes are part of the filename. When SEISAN looks for a SEED response
files it will search for a file with a name matching the network, station, location and channel code of the
data in memory.

The RESP filename must match the waveform data, so if the waveform data does not include network
code and location code, then filename must be e.g.: RESP..BER..BHZ

So if the location code in the waveform data is ‘–’ then the RESP filename must include the ‘–’, like e.g.:
RESP.NO.BER.--.BHZ

And if the location code in the waveform data is empty ‘ ’ then the RESP filename must be like: RESP.DK.BSD..BHZ

The SEED response files can include multiple changes of response and must have valid Start and End
dates for the time of the data that are processed. For the current response the ”End date” must be set
to ”No Ending Time”.

SEISAN will use this scheme until it finds a valid response file:

1: Look for SEED response file in the current working directory.
2: Look for SEED response file in the CAL directory.
3: Look for a SEISAN, GSE or SAC response file in the current working directory.
4: SEISAN will check if the environmental variable LOCAL_CAL is set, if it is, SEISAN will look for
   a SEISAN, GSE or SAC response file in the LOCAL_CAL directory.
5: Look for a SEISAN, GSE or SAC response file in the CAL directory.
6: Look for a SEISAN, GSE or SAC response file in the CAL/STAT directory.

In the very unlikely case where a selected data window covers two different response files, the response
file belonging to the first part of the data is used. In the very unlikely case where a selected data window
covers two different response files, the response file belonging to the first part of the data is used.

The RESP program (section 41) can be used to generate the response files. The filenames for the response
files are STATTCOMP.YYYY-MM-DD-hhmm FOR where STATT is station code, COMP is component,
YYYY is year, MM is month, DD is day, hh is hour, mm is minute and FOR is the format indicator
which can be SEI or GSE. If FOR is not given, the format is SEISAN. An example is BER_S.Z.1999-
05-05-1244.. You should take a backup of the response files before you run the program (see chapter 3).
The response files can be located in CAL, or, if many files are available optionally also in a subdirectory
structure. This optional structure simply consist of a subdirectory for each station and the subdirectory
name must have 5 letters so base BER would have the name BER.. The system automatically locates
the response files whether all are in CAL or in the subdirectory structure.

The response file can store the response in different ways:
1 SEISAN format:
   a Parameters used for calculating the response: Generator constant, filters etc. In addition, the response (amplitude and phase) at 30 frequencies is listed. In this case the response is calculated from the parameters.
   b Incomplete set of parameters or no parameters and the response at 30 frequencies. In this case the response is calculated by interpolation of the 30 values.
   c Poles and zeros: No discrete values are given and the response is calculated directly from the poles and zeros. The number of poles and zeros in the SEISAN format is limited.

2 GSE CAL2 format:
   a Poles and zeros, number is unlimited, the response is directly calculated from the poles and zeros.
   b Pairs of frequency amplitude phase, number of pairs is unlimited, the response is calculated by interpolation.

3 SEED
   Poles and zeros (only Transfer function type A (Laplace Transform (Rad/sec)), number of poles and zeros are unlimited, the response is directly calculated from the poles and zeros. Only reads SEED response in ASCII format as written out by rdseed, not dataless SEED volumes. The command to extract the files is `rdseed -R -f seed_filename`. Standard filename such as `RESP.1U.TRIS.10.BHZ` are understood. Files are read from CAL directory. Note that if e.g. the horizontal channels are rotated the `-R` flag by rdseed will not provide this information.

4 SAC format: SEISAN can use SAC PAZ files as created by rdseed. The files have to be names in the standard SEISAN way, but have to end on _SAC.

NOTE: When rotating signals, it is in SEISAN assumed that the response is the same on all 3 channels. !!! Response files can be plotted from MULPLT showing the actual response information that is used with a given trace. Response files can also be plotted directly with program PRESP, see below. All or a subset of the response files can be printed out in a table with program PRRESP. The program must be executed from the directory with the response files. Make a listing (file filenr.lis) of files to print out with DIRF and run the program. It will produce an output file ready for printing. A response file can be plotted with the program PRESP. The program is started with command `presp filename`, where `filename` is the response file name. If no file name is given, the program asks for a filename or number. If a DIRF has been made and the list of files in filenr.lis is available, a response file can then be selected with a number. The program produces a PostScript output file with name `presp.eps`.

41.1 Create instrument response files, RESP

Introduction
The purpose of this program is to (1) Make Seisan or GSE2 response files, (2) Provide the engineer maintaining seismic instrumentation with a practical tool for calculating and checking response functions of the most common elements of a seismic system. The program can calculate response functions of velocity transducers, accelerometers, filters and amplifiers, input poles and zeros or tabulated values and multiply the combinations together to get complete system response functions. The program produces a table with the response function and a simple graphical expression of the response curve. For the purpose
of checking measured values, a file with these values can be used as input and will be plotted together with the theoretical values. The program can calculate, acceleration, velocity or displacement response. Program PRRES can make a table of many response files.

**The instrument response**

The seismic recording system can consist of seismic sensor, analog-digital converter, amplifier and filters. For a detailed discussion the user is referred to Scherbaum [1996]. The combined response can be given in the frequency domain as frequency response function or in the Laplace domain as transfer function. The frequency response is given in pairs of frequency amplitude phase (FAP), while the transfer function is given as poles and zeros (PAZ). The combined frequency response is obtained through multiplication of the response from the individual components, while the transfer function is obtained by combing the PAZ from the components. Amplifiers and accelerometers are specified simply by a constant gain. Filters are assumed to be Butterworth. RESP can be used to write finite impulse response (FIR) coefficients [Scherbaum, 1996] that are used as anti-alias filters in most modern digitizers if GSE is used as output format. SEISAN has no capability to read the FIR filters or to correct for them. However, the FIR filters are part of a full description of the instrument response and should be at least included for information if possible.

The electrodynamic seismometer is assumed to have the following velocity frequency response:

\[ T(\omega) = \frac{\omega^2}{\omega_0^2 - \omega^2 + i2\omega\omega_0h} \]

which corresponds to the transfer function:

\[ T(s) = \frac{-s^2}{\omega_0^2 + s^2 + 2s\omega_0h} \]

where \( s = i\omega \), \( \omega \) is the angular frequency \( 2\pi f \) in Hz, \( \omega_0 \) the resonance frequency of the seismometer, \( i = \sqrt{-1} \) and \( h \) the damping (normally around 0.7).

**NOTE:** In the equation for the frequency response, the sign "+ 2*i*h.." was "-" before March 2000, so old parameter files may have to be regenerated. The sign depends on the definition of the signs in the Fourier transform and therefore may be different in different text books. It may even be wrong although it looks right, if a wrong Ansatz is done. Due to the wrong sign, the FAP values in the SEISAN response files were wrong, however the programs use the constants given in the files and the correct response is generated. If you have the instrument constants in your old response files and not just FAP, the old response files can be used.

The transformation from displacement to velocity or back is done by multiplying with \( i^m \).

In addition to or instead of using the equation above, values can also be entered as discrete values or as poles and zeros.

The SEISAN response function is calculated for 60 frequencies between 0.01 and 100 Hz and the steps between the frequencies are approximately logarithmic. The response function is normalised at 1.0 Hz (see Table 1) and the gain at 1.0 Hz is given separately.

**NOTE:** ALL UNITS ARE IN METERS, SECONDS OR G (9.8ms-2)

**NOTE:** It seems that although the GSE format is clearly defined, there has been different interpretations. This has also led to changes in SEISAN since the GSE response was introduced with SEISAN. For more details, see Appendix C.
41.1. CREATE INSTRUMENT RESPONSE FILES, RESP

Which format to use

SEISAN, since version 7.1, supports the GSE2 calibration format in addition to the SEISAN response file format. We recommend that you use the GSE2 format, since it presents one of the most widely used calibration formats. Storage of the response in terms of PAZ is recommended over FAP, since the PAZ representation describes the continuous transfer function. You may continue using existing SEISAN response files and add new files in GSE2 format, or replace the old SEISAN response files with new GSE2 files.

How to run the program

The program has quite a few options, which easily may lead to confusion. Before you start you should know which format you want to use (GSE2 or SEISAN) and whether you want to describe the response in terms of FAP or PAZ. The recommended choice is to use GSE2 and PAZ.

Type RESP to start the program. You will then get a series of questions as indicated below in upper case letters. All input is format free. A sample run is shown below.

CHOSE OUTPUT FORMAT: 0: NO OUTPUT FILE
1: SEISAN FAP
2: SEISAN PAZ
3: GSE2 FAP
4: GSE2 PAZ

Answer with 0-4, options 1-4 will create respective response files in selected format, option 0 will only calculate and show the response on the screen. SEISAN PAZ can only be used if number of poles and number of zeros are less than 38. If more are input, a table will be generated automatically in FAP format. A format with poles and zeros MUST BE USED if a mechanical displacement sensor is selected or an accelerometer is selected which does not have first character A in component name (see later).

TYPE OF SENSOR: 1: NONE
2: SEISMOMETER
3: ACCELEROMETER
4: MECHANICAL

DISPLACEMENT SEISMOMETER

Answer with 1, 2, 3 or 4. Number 1 is used when only calculation of filters or amplifiers are desired, 2 is a standard velocity transducer, 3 a standard accelerometer and 4 is a mechanical displacement sensor (normally a digitized record of a paper seismogram). If a seismic sensor is used, you will get additional questions on the constants of the sensor. If a seismometer is chosen, the following questions must be answered:

SEISMOMETER NATURAL PERIOD ?
This is measured in seconds. For most short period systems the value would be 1.0 second.

SEISMOMETER DAMPING RATIO ?
The damping ratio should ideally be 0.7. This depends on the damping resistance.

For both the electrodynamic seismometer and accelerometer, the following question is given:

SENSOR LOADED GENERATOR CONSTANT (V/M/S OR V/G) ?
This is the generator constant of the sensor in terms of volt per unit of of ground motion (meter/second or g). It is important to note that this is the loaded constant, which means the effective output of the sensor taking into account amplifier input and damping resistances.

For the mechanical sensor, the question is
SEISMOGRAPH GAIN in TIMES GAIN (M/M) ?
which is simply the mechanical gain.

Now comes questions about amplifier, filter and recording unit.

RECORDING MEDIA GAIN (COUNT/V OR M/V) ?
If you have a recording media, the gain can be given here, otherwise just enter 1.0

If the output format is GSE, the response is always calculated in displacement units, while for SEISAN output and seismometer or accelerometer, the following options appear:

TYPE OF RESPONSE: 
1: DISPLACEMENT
2: VELOCITY
3: ACCELERATION

Normally for a seismometer, one wants to calculate the displacement response and for an accelerometer, the acceleration response. However it might sometimes be interesting to look at e.g. the velocity response for a seismometer (after all, the seismometer is normally a velocity transducer !!). Enter the appropriate number.

AMPLIFIER GAIN (DB) ?
This is the amplifier gain in dB. Since this question is only asked once, this gain must include gain of all units except the recorder (asked below). This could e.g. include gain of the VCO system.

NUMBER OF FILTERS (0-10), RETURN FOR NONE ?
Up to 10 filters can be specified. If you answer 0, no filters are used and no more questions on filters will appear. Otherwise one line of input must be given for each filter as follows:

FREQUENCY AND NUMBER OF POLES FOR EACH FILTER,
POLES NEGATIVE FOR HIGH PASS

Each line requires two numbers, the corner frequency of the filter and the number of poles. A high pass filter is given by letting number of poles be negative. It is not always easy to know whether a filter is e.g. one 2 pole or two 1 pole filters, the user needs to experiment with this.

FILE NAME FOR FILE WITH POLES AND ZEROS, RETURN FOR NO FILE
Here a file with poles and zeros can be entered. If seismometer constants have been chosen above, the values calculated with poles and zeros are multiplied with the values previously calculated. The free format file contains:

1. line: NP: Number of poles, NZ: Number of zeros, Norm: Normalization constant
   Following NP lines contain one pair each of real and imaginary poles
   Following NZ lines contain one pair each of real and imaginary zeros

NOTE: The unit of frequency is radian/s so if in Hz, multiply with $2\pi$ and normalization constant in $\text{radian} = \frac{\text{normalization constant in Hz}^2 \pi}{\text{number of poles} - \text{number of zeroes}}$.

The next 2 options are only shown if the output file is selected to be FAP:

FILE NAME FOR TABULATED VALUES, RETURN FOR NO FILE
Here a file with tabulated values are entered. If seismometer constants or poles and zeros have been chosen above, the tabulated values will be interpolated and multiplied with the values previously calculated for from above. The free format file contains:

1. line: N: Number of tabulated values, Norm: Normalization constant
Following N lines contain one each frequency, amplitude and phase (deg).

GIVE FILE NAME FOR MEASURED VALUES, RETURN FOR NONE

Give file name for measured values. In most cases you have none so just make a return. The format of the input file is as follows:

- frequency, amplitude, phase
- frequency, amplitude, phase
- etc.

E.g.

0.2,0.7,200
0.7,0.8,100
10.0,0.1,33

The file has no blank lines and can contain up to 60 data sets. It is important to note that the amplitude values should be NORMALIZED at 1.0 Hz.

Now there is no more input to the response parameters, and the output is:

GAIN FACTOR AT 1.0 HZ: 12345.6

This is the gain of the system at 1.0 Hz and is also the value for normalizing the response curve, that is, all calculated values are divided by this number. There is no unit for gain of an amplifier and for displacement response using a seismometer and drum recording. If the recording is digital, the unit would be counts/meter and for a velocity response counts/meter/second etc. If a file with poles and zeros is used without any other information, the normalization constant must have the unit of count/m, similar for the tabulated input.

Further output is given in a file called resp.out, see Table 1 for an example.

The response curves (amplitude and phase) are now printed/plotted on the screen. First comes the amplitude response (amplitude in db versus log frequency). By pushing return, the phase response is shown (phase shift (deg) versus log frequency). After the plots, the SEISAN calibration file can optionally be made, follow instructions, see example below. The response file MUST be calculated for the displacement response, and all calculation in SEISAN assume that response is calculated in counts/m.

After the SEISAN response file is made, the current parameters will be displayed and one or several can be changed without entering all again. Like if the gain has changed at a certain date, only change date and gain. This feature (new in SEISAN7.2) has been put in to be able to quickly make many similar response files, like when all files have to be put in for a network.

**Comments to data for response files**

Station and channel codes

It is important that the station and channel codes are made exactly as they appear in the waveform files. If not, SEISAN is not able to identify the channel.

Date

The date given here corresponds to the date from which the calibration information is valid. The SEISAN system will always look for the most recent calibration file relative to the date of the earthquake.

Latitude, longitude and elevation

These data are for information only, it is not used anywhere in SEISAN, so it does not have to be entered, however there is room for it in the SEISAN waveform file headers.
Comment
No information used by the system.

Plot
After the response file has been written out, a plot is made with PRESP of the file. There will also be a
plotfile, *presp.eps*, which can be sent to the printer. The response file can store the response in different
ways:

1. Parameters used for calculating the response: Generator constant, filters etc. In addition, the
response (amplitude and phase) at 30 frequencies are listed. In this case the response is calculated
from the parameters.

2. Incomplete set of parameters or no parameters and the response at 30 frequencies. In this case the
response is calculated by interpolation of the 30 values.

3. Poles and zeros: No discrete values are given and the response is calculated directly from the poles
and zeros.

See also Appendix B for the SEISAN waveform file format and section 5.2.2.

**IMPORTANT:** PUT RESPONSE FILE IN CAL DIRECTORY OR ONE OF ITS STA-
TION SUBDIRECTORIES. Response files can also be in working directory but this is not
advisable except for testing.

Example of running the program:

```
oxygen-(larso)23: resp

RESP - PROGRAM TO CREATE RESPONSE FILES IN SEISAN
OR GSE FORMAT. THE RESPONSE CAN BE CREATED
AS POLES AND ZEROS (PAZ) OR FREQUENCY
AMPLITUDE AND PHASE (FAP). THE SAME
TRANSFER FUNCTION AND FILTERS ARE USED
IN BOTH CASES.

CHOOSE OUTPUT FORMAT: 0: NO OUTPUT FILE
1: SEISAN FAP
2: SEISAN PAZ
3: GSE2 FAP
4: GSE2 PAZ

4

TYPE OF SENSOR: 1: NONE
2: SEISOMETER
3: ACCELEROMETER

2

SEISOMETER NATURAL PERIOD ?
1

SEISOMETER DAMPING RATIO ?
.7

SENSOR LOADED GENERATOR CONSTANT (V/M/S OR V/G) ?
300
```
41.1. CREATE INSTRUMENT RESPONSE FILES, RESP

RECORDING MEDIA GAIN (COUNT/V OR M/V) ?
2048
AMPLIFIER GAIN (DB) ?
40
NUMBER OF FILTERS (0-10), RETURN FOR NONE ?
1
FREQUENCY AND NUMBER OF POLES FOR EACH FILTER,
POLES NEGATIVE FOR HIGH PASS
10 2
FILE NAME FOR FILE WITH POLES AND ZEROS, RETURN FOR NO FILE

FILE NAME FOR MEASURED VALUES, RETURN FOR NO FILE

<table>
<thead>
<tr>
<th>AMPL</th>
<th>AMPLITUDE RESPONSE</th>
<th>SEISMOMETER</th>
<th>DISPLACEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.88</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>3.66</td>
<td>.</td>
<td>++</td>
<td>.</td>
</tr>
<tr>
<td>1.36</td>
<td>.</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>0.504</td>
<td>.</td>
<td>+</td>
<td>++</td>
</tr>
<tr>
<td>0.187</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>0.694E-01</td>
<td>.</td>
<td>++</td>
<td>.</td>
</tr>
<tr>
<td>0.257E-01</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>0.955E-02</td>
<td>.</td>
<td>++</td>
<td>.</td>
</tr>
<tr>
<td>0.354E-02</td>
<td>.</td>
<td>+</td>
<td>.</td>
</tr>
<tr>
<td>0.131E-02</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>0.488E-03</td>
<td>.</td>
<td>++</td>
<td>.</td>
</tr>
<tr>
<td>0.181E-03</td>
<td>.</td>
<td></td>
<td>.</td>
</tr>
<tr>
<td>0.671E-04</td>
<td>.</td>
<td>++</td>
<td>.</td>
</tr>
<tr>
<td>0.249E-04</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>0.924E-05</td>
<td>.</td>
<td>++</td>
<td>.</td>
</tr>
<tr>
<td>0.343E-05</td>
<td>.</td>
<td>++</td>
<td>.</td>
</tr>
<tr>
<td>0.127E-05</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>0.472E-06</td>
<td>.</td>
<td>++</td>
<td>.</td>
</tr>
<tr>
<td>0.175E-06</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

----------------------------------------------------------------------------------
| FREQ  | 0.01  | 0.03  | 0.14  | 0.71  | 3.68  | 19.19 | 100.00 |
| GAIN FACTOR AT 1 HZ: 0.276E+09 RETURN FOR PHASE RESPONSE |

<table>
<thead>
<tr>
<th>PHAS DEG</th>
<th>PHASE RESPONSE</th>
<th>SEISMOMETER</th>
<th>DISPLACEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>163.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>144.</td>
<td>.</td>
<td>++</td>
<td>.</td>
</tr>
<tr>
<td>125.</td>
<td>.</td>
<td>++</td>
<td>.</td>
</tr>
<tr>
<td>106.</td>
<td>.</td>
<td>++</td>
<td>.</td>
</tr>
<tr>
<td>87.1</td>
<td>.</td>
<td>++</td>
<td>.</td>
</tr>
<tr>
<td>68.0</td>
<td>.</td>
<td>++</td>
<td>.</td>
</tr>
</tbody>
</table>
CHAPTER 41. INSTRUMENT RESPONSE

<table>
<thead>
<tr>
<th>FREQ</th>
<th>0.01</th>
<th>0.03</th>
<th>0.14</th>
<th>0.71</th>
<th>3.68</th>
<th>19.19</th>
<th>100.00</th>
</tr>
</thead>
</table>

GSE RESPONSE FILE (Y/N=default)?y

Enter station code. e.g. BERGE, max 5 chars

TEST

Enter component (4 chars) e.g. SH Z
First character is type, should be one of the following:
E, S, H, B, M, L, U
Second character is instrument code, should be one of the following:
H, L, G, M, N
Third character is not used, leave blank
Last character is orientation, must be Z, N or E

S Z

Enter date as YYYYMMDDHHMMSS, at least up to the day (e.g. 19880123):20000101
Latitude (Up to 4 decimal places and - for south), return for none:
Longitude (Up to 4 decimal places and - for west), return for none:
Enter elevation in meters (integer), return for none:
Enter comments, one line. e.g. amp type, sensor type return for none

Response file name is: TEST_S__Z.2000-01-01-0000_GSE

RESPONSE CURVE IS IN FILE resp.out

Problem: The response file naming has not changed according to the SEED convention so location and network code cannot be entered using RESP, however they can be entered manually in response file (see response file format). This means that e.g. SHZ must be entered as SH Z since only 3 of the 4 character are used. For older data in SEISAN format, all 4 characters can be used.

Examples of response files is given in Appendix C.
41.2 Examples of main response files from seismometers and accelerometer

Example of a Güralp DM 24 digitizer with CMG-40-1 (1 Hz)

**Digitizer:**

The sensitivity of the digitizer is given to $3.197 \mu V/count$. The SEISAN gain is in $\text{counts/V}$ so

$$\text{SEISAN recording media gain} = \frac{1000000}{3.197} = 312793 \text{ count/V}$$

**Sensor:**

Sensitivity is $2 \times 1001 \text{ V/m/s} = 2002 \text{ V/m/s}$

**Making response file with parameters**

For calculating with parameters, it is assumed that the free period is 1.0 s and damping is 0.7. Using the `resp` program answering as follows:

- Output format: 0 (Only testing)
- Type of sensor: 1 (It is a seismometer)
- Seismometer period: 1.0
- Seismometer damping: 0.7
- Generator constant: 2002
- Recording media gain: 312793
- Amplifier gain: 0 (No amplifier)
- Number of filters: enter (No filter)
- File with poles and zeroes: enter
- File with tabulated values: enter
- File with measured values enter

Then the plot below comes up.

![Figure 41.1: Making response file with parameters.](image)
Making response file with poles and zeros

The poles and zeroes velocity response in units of Hz is given as

Poles
-0.707 0.707
-0.707 -0.707
-62.4 135.4
-62.4 -135.4
-350.0 0.0
-75.0 0.0

Zeros
0.0 0.0
0.0 0.0

SEISAN units are radians/sec so poles and zero values are multiplied by $2\pi$.
The normalization constant is given as $585.810^6$. To convert to radian is done as follows

Normalization constant in radian = $585.810^6(2\pi)(\text{number of poles} - \text{number of zeroes}) = 585.810^6(2\pi)^4 = 9.1210^{11}$.

SEISAN also uses displacement so one zero is added. The values are then

Poles
-4.442 4.442
-4.442 -4.442
-392.0 850.7
-392.0 -850.7
-2199.0 0.0
-475.0 0.0

Zeros
0.0 0.0
0.0 0.0
0.0 0.0

To get total constant (gain and normalization constant), we multiply by sensor gain and digitizer gain

Total normalization constant = $9.1210^{11} \times 2002 \times 312793 = 5.7110^{20}$

A SEISAN input file is then made

6 3 5.71e20 6 poles, 3 zeros and total gain constant
-4.442 4.442
-4.442 -4.442
-392.0 850.7
-392.0 -850.7
-2199.0 0.0
-475.0 0.0
0.0 0.0
0.0 0.0
0.0 0.0

The resp program now makes the SEISAN response file with this input as follows

Output format: 0 Only testing
Type of sensor: 0 Sensor response is in poles and zero file
41.2. EXAMPLES OF MAIN RESPONSE FILES FROM SEISMOMETERS AND ACCELEROMETER

![Figure 41.2: Making response file with poles and zeros.](image)

Recording media gain: 1 Gain has been put into total gain constant
Amplifier gain: 0 No amplifier
Number of filters: enter No filter
File with poles and zeroes: resp.inp File with poles and zeros, can be any name
File with tabulated values: enter
File with measured values enter

Then the plot below comes up

It is seen that the two ways of making the response file gives almost the same result, however using poles and zeroes is the most accurate, particularly for active sensors. In both cases no consideration was made for antialias filters which normally can be disregarded if a modern sharp filter.

**Example of a Gurlp DM 24 digitizer with CMG-5T accelerometer**

The digitizer is the same as before

Using parameter format, SEISAN currently requires the component name to start with A. According to international standards, the component code for an accelerometer should be something like ENZ so a parameter format cannot be used if international coes are used and poles and zeroes must be used.

For the CMG-5T, the only information about the sensor is the sensitivity of 1V which is is equivalent to 0.970 m/s² or 1.03 V/ms². In SEISAN parameter format this should be converted to V/g so sensitivity is then

\[9.81 \text{ (ms}^{-2}/\text{g})/0.97(\text{ms}^{-2}/\text{V}) = 10.1 \text{ V/g}\]

**Parameter format**

The input is:

Output format: 1 parameter format
Type of sensor: 3 It is an accelerometer
Generator constant: 10.1
CHAPTER 41. INSTRUMENT RESPONSE

Figure 41.3: Making response file for an accelerometer with parameters.

Recording media gain: 312793
Amplifier gain: 0 No amplifier
Number of filters: enter No filter
File with poles and zeroes: enter We use parameters now
File with tabulated values: enter
File with measured values enter

The plot below comes up

A response file can now be made with e.g. A Z as component code.

**Poles and zeros**

The displacement response for an accelerometer consists of 2 zeros and normalization constant of 1. The total gain constant is then

312793 x 1.03 = 322000

The generator constant is in V/ms-2 since the two zeros transform from acceleration to displacement.

So the input file e.g. resp.txt with poles and zeros is

0 2 322000
0 0
0 0

The manual input is

Output format: 2 PAZ in SEISAN format

Type of sensor: 1 It is give input resp file

Generator constant: 1.0 Given in resp input file
Recording media gain: 1.0 Given in resp input file

Amplifier gain: 0 No amplifier

Number of filters: enter No filter

File with poles and zeroes: resp.txt Use resp.txt input file

File with tabulated values: enter

File with measured values enter

The rest is the same

The response file would then be

```
TEST EN Z100 0 0 0 0 0.000 P
0 2 0.3220E+06 0.000 0.000 0.000 0.000
```

Making a response file for a particular station

For a particular station, chose output format SEISAN PAZ or GSE2 PAZ and later answering yes to question of making the SEISAN response file (see SEISAN manual ???????????????). If e.g. the station has station code TEST and component name S Z, the a response file valid from January 1, 2007 will have the name TEST_S..Z.2007-01-00-0000.ISE. In case of a SEISAN poles and zero file, the content is:

```
TEST S Z107 0 1 0 0 0 0.000 P
6 3 0.5710E+21 -4.442 4.442 -4.442 -4.442 -392.0
850.7 -392.0 -850.7 -2199. 0.000 -475.0 0.000
0.000 0.000 0.000 0.000 0.000 0.000 0.000
```

So the file could have been made without using resp.

Table 1 Example of resp.out:
### 41.3 SEED response

SEISAN can directly read SEED responses, which is poles and zeros, given as velocity response and transfer function types A (Laplace Transform in Rad/sec) and B (Analogue in 1/sec). Storage of response in one of these is the most common. The resp files can be created with rdseed from a full or dataless SEED volume (`rdseed -R -f seed_volume`). RDSEED creates files with the pattern `RESP.NC.STAT.LC.CHC`, where NC=network code, STAT=station code, LC=location code (not used by SEISAN) and CHC=channel code. The resp files need to be stored in the CAL directory and SEISAN will find the correct file. The RESP filename must match the waveform data, so if the waveform data does not include network code and location code, then filename must be e.g.: `RESP..BER..BHZ`

So if the location code in the waveform data is `--` then the RESP filename must include the `--`, like e.g.: `RESP.NO.BER.--.BHZ`

And if the location code in the waveform data is empty ` ` then the RESP filename must be like: `RESP.DK.BSD..BHZ`

If there are many files to change, program `CHANGE_RESP_NAME` can be used:

- make a dirf of RESP files that all require the same change to location and network
- length of station name is unimportant
- questions of changing/adding location and/or network codes, to see codes, plot with MULPLT

---

FOR MORE DETAILS ON HOW TO UNDERSTAND GSE AND SEED RESPONSE PARAMETERS, SEE [Havskov and Alguacil, 2004], chapter 6.
41.4 SEED RESPONSE TO GSE, SEEDRESP2GSE

- if file has one '.' for code, it will be replaced by '

The resp file can contain response information from several time intervals. SEISAN uses the date and time of the waveform data to find the corresponding instrument response.

SEED response files are given in stages, for example seismometer, digitizer and FIR filters are stored as individual stages. The overall response is made by combining all the stages. SEISAN uses the following blockets from the SEED resp file (for more details see [IRIS Consortium 1993]):

- B052F22 - start date
- B052F23 - end date
- B053F03 - transfer function type, A=Laplace Transform (Rad/sec), B=Analog (1/sec)
- B053F07 - A0 normalization factor (A0 is checked against poles and zeros at normalization frequency and changed if not correct). The product of poles and zeros at the normalization frequency and A0 gives 1.
- B053F08 - Normalization frequency
- B053F10-13 - zeros, if transfer function type is B, normalization factor A0 is changed to (A0)/(2 pi) for each zero
- B053F15-18 - poles, if transfer function type is B, normalization factor A0 is changed to (A0)*(2 pi) for each pole
- B058F04 - gain

The overall gain factor is given by the product of normalization factors and gain factors from all stages. One zero is added to convert to displacement response. It is assumed that input units are V/m and output units are counts, no checks are done on input and output units.

41.4 SEED response to GSE, SEEDRESP2GSE

SEEDRESP2GSE converts SEED resp files as written out by rdseed to GSE format. The program only supports poles and zeroes and transfer function type Laplace Transform. The program asks for station and component names and a time. This is because the resp file could have data from several channels and cover several time intervals with different instrument configuration.

41.5 GSE response to SEED, GSERESP2SEED

GSERESP2SEED can be used to build dataless SEED volumes from a set of GSE calibration files. The conversion is based on the GSE2SEED program by Reinould Sleeman (email sleeman@knmi.nl). Input can be single filenames or a list of files given in fileenr.lins. The program produces a single channel SEED volume for each channel given by a GSE response file. At the end of the output filename, GSE is replaced by SEED. Other tools have to be used to merge several channels into one SEED volume. If there are several GSE files for a channel from different time periods, a stop date has to be given in the CAL2 line of the respective GSE file. Station coordinates are taken from the STATION0.HYP file. The program can use the site name, if it is part of the GSE response file through a comment line as in the following example:

(GSE2SEED_SITENAME Charnwood Forest, Leicestershire, England, UK)
Chapter 42

Macroseismic data in SEISAN

Macroseismic information in SEISAN are of 2 kinds. The summary information with maximum intensity, macroseismic epicenter etc has a special line in the S-file (see Appendix1) and the SELECT program can search for felt events. In addition, from SEISAN, version 8.1, a format has been defined to store macroseismic observations used to create e.g. maps with isoseismals. The observations files are stored in a local ISO format. For a format description and suggestion for file names, see Appendix A. There are currently no SEISAN programs that generates these files so they have to be made manually from the observations. The files in ISO are linked to the events as given by the S-file data base structure in the same way as waveform files are linked to the events. The line type is MACRO3 and an example is

```
2007-01-21-1345-00.MACRO MACRO3
```

Thus information about event source parameters and felt information is available together. An example of a file is

```
Brattvag, More og Romsdal  2007  121 1345  0 GMT  2007  121 1445  0 Local time
Comment
62.200  6.110  5.0 EMS  6150 ARSTA
62.200  6.110  4.0 EMS  6150 ARSTA
62.890  7.680  4.0 EMS  6480 BATNFJORDSARA
62.500  6.680  6.0 EMS  6260 SKODJE
62.460  6.130  6.0 EMS  6013 Alesund
62.200  6.110  6.0 EMS  6150 ARSTA
62.460  6.290  6.0 EMS  6017 ASESTRANDA
62.460  6.010  4.0 EMS  6055 GODAY
62.490  6.176  5.0 EMS  6057 ELLINGSAY
62.430  6.180  4.0 EMS  6030 LANGEVAG
```

The file format is given in Appendix A. Program EPIMAP can plot the new files (use macroseismic file instead of a hypocenter file). The requirement is that the the first 3 letters after the ‘.’ is mac or MAC (as example above). The intensities will be plotted as number on the map. A new Unix program can also be used with the data. Program MACROMAP can use the macroseismic observation file as input to
create a map of the observations using GMT. The program generates a GMT script file, macromap.gmt, which then is executed from within the program to create a PostScript output file, macromap.ps. This file is then displayed, from within the program, with Unix command gv (GhostView). The program also runs under Windows, but does not plot.

The input can also be from a file made with macroquest (web based interactive program for input from the public, to be distributed with SEISAN CD). In addition to making the plot, a conversion from the web format to SEISAN format is made (output file macromap.out). This option requires an input file with postal codes in order to get location of the observations. MACROMAP can also be executed directly or from eev. When executed directly from the prompt line, the options are: -macroinput file with macroseismic observations, SEISAN format, abs path or in ISO -placename optional additional file with place names, to be shown on map, abs path or in DAT, epimap format is used -postfile optional file with postal code, abs path, used with web option If used with eev, the place name file must have name place_names.macro

An example of the postal code file is

<table>
<thead>
<tr>
<th>Brattvag, More og Romsdal</th>
<th>2007 121 1345 0 GMT 2007 121 1445 0 Local time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comment</td>
<td></td>
</tr>
<tr>
<td>Brattvag, More og Romsdal</td>
<td></td>
</tr>
<tr>
<td>62.200</td>
<td>6.110 5.0 EMS 6150 ARSTA</td>
</tr>
<tr>
<td>62.200</td>
<td>6.110 4.0 EMS 6150 ARSTA</td>
</tr>
<tr>
<td>62.890</td>
<td>7.680 4.0 EMS 6480 BATHFJORDSARA</td>
</tr>
<tr>
<td>62.500</td>
<td>6.680 6.0 EMS 6260 SKODJE</td>
</tr>
<tr>
<td>62.460</td>
<td>6.130 6.0 EMS 6013 Alesund</td>
</tr>
<tr>
<td>62.200</td>
<td>6.110 6.0 EMS 6150 ARSTA</td>
</tr>
<tr>
<td>62.460</td>
<td>6.290 6.0 EMS 6017 ASESTRANDA</td>
</tr>
<tr>
<td>62.460</td>
<td>6.010 4.0 EMS 6055 GODAY</td>
</tr>
<tr>
<td>62.490</td>
<td>6.176 5.0 EMS 6057 ELLINGSAY</td>
</tr>
<tr>
<td>62.430</td>
<td>6.180 4.0 EMS 6030 LANGEVAG</td>
</tr>
</tbody>
</table>

The content is postal code, latitude, longitude and location. The format is a10,ff10.3,2x,a30. the postal code does not have to be a number, but can be any string.
Figure 42.1: Macroseismic map made with MACROMAP using EEV. The epicenter, taken from the S-file, is shown with the star.
Chapter 43

Correlation of waveform signals, CORR and detection of event clusters XCLUST

The cross-correlation function provides a measure of similarity between signals. In seismology cross-correlation can be used to measure the similarity of waveforms between seismic events and in case of similarity to determine relative arrival times of a seismic wave between two events. Waveform similarity is caused by proximity in hypocenter location and similarity in focal mechanism between two events. Cross correlation can be computed with the program CORR and the output be processed with program XCLUST to detect groups of similar events.

**CORR**

The program CORR computes correlation and can be used to measure relative arrival times. It also can be used to determine correlation of a master event with continuous data and extract event based data. The output of maximum correlation for a station recording different events can be used to identify event clusters within the data set, this can be done using the program XCLUST. The cross-correlation function of signals x and y is computed in the time domain as:

\[ r_{xy}(i) = \frac{\sum_{j=1}^{n} x_j y_{(j+i-1)}}{\sqrt{\sum_{j=1}^{n} x_j^2} \sqrt{\sum_{j=1}^{n} y_j^2} + i - 1} \]

Phase arrivals

Phase arrivals of similar events can be determined accurately through cross-correlation. P and S arrivals can be determined independently. The procedure starts by selecting and picking phases for a master event that is representative for the group of events. Analysis of this event needs to be done accurately as it is the basis for the subsequent analysis. Phases of the other events in the group can be determined through cross-correlation of either the complete trace or a selected phase window with the master event. To pre-select a time window, manual identification of the phase for subsequent events is required prior to running CORR. This may be necessary for example if the waveform file contains several events. The phase arrival time is given by the maximum of the cross-correlation function that needs to exceed a minimum threshold. The arrival time is written out as absolute time. Filtering is applied to the signals
if selected for both master and subsequent events, this may be necessary especially when dealing with events of different size. The filtering introduces a phase shift, which is applied to both signals. However, the absolute phase arrival for the subsequent event is consistent with the master event picked time.

The calculation of the relative phase time (\(dt\)) is done by taking the travel time for the master event (\(AT1 - OT1\)) (where \(AT\) is arrival time and \(OT\) origin time) minus the travel time of the other event (\(CAT - OT2\)), where \(CAT\) is the time corresponding to the maximum amplitude in the cross correlation function. The output file \(dt.cc\) can be used with the double difference location program HYPODD.

**Cross correlation matrix**

In this mode the cross-correlation is computed between the same stations for all pairs of events (that fulfill the criteria for maximum distance between the events, and event and station). The resulting cross-correlation matrix (for each station containing \(\sum_i^n (i - 1)\) values, where \(n\) is the number of events) can be used to identify groups of similar events using the program XCLUST.

**Continuous mode**

The main objective of running CORR in this mode is to identify a master waveform signal in a continuous data stream, given by waveform data files. The times when correlation is higher than the selected threshold level are written out, and can be visualized by splitting the corr.out file and using EEV and MULPLT. In addition, it is possible to cut out individual event files (see CONTINUOUS EXTRACT parameter).

**Input file**

Input to the program is given through the file corr.inp. A sample file is given in the DAT directory; the data used in the example are part of the test data set (TEST database 2003/06, see training document). The program is run by command corr in the same directory as corr.inp and the s-files. The waveform files can be in any SEISAN standard place. All standard waveform formats can be used.

The parameters in **corr.inp** are as follows:

**Event file names:**

- **SFILE MASTER:** sfile name of master event, remove or comment out this parameter to run program in group identification mode to determine cross-correlation matrix between all events and identify group of similar events
- **SFILE EVENT:** sfile name of events that will be either cross-correlated among themselves, or compared to the master event, there can be several of these.
- **SFILE INDEXFILE:** textttfilenr.lis file can be used to give S-file names instead of listing them with ‘SFILE EVENT’

**General parameters:**
INTERACTIVE: set to 1. for interactive use where graphics are displayed on the screen, which is useful for testing; or set to 0. for non-interactive run.

CC MATRIX WINDOW: time interval in seconds to use in computation of cross-correlation matrix instead of the duration given for each station given on STATION line, if different from 0.; full trace is used if EVENT SELCRIT is set to 0.

CONTINUOUS MODE: write out all detection times for correlation above threshold if set to 1.; otherwise only phase for maximum correlation CONTINUOUS EXTRACT: extract event waveform files for correlation above threshold, 0. for no extract, 1. to extract single channel used, 2. extract all channels

EVENT SELCRIT: cross-correlation with the master signal can be computed either for the complete trace of the subsequent event (0.) or the same part of the signal (either P or S) as for the master event (1.) including the pre signal part and of the same duration as defined in STATION line

FILTER: this parameter allows to enable (1.) or disable (0.) filtering as defined for each channel with parameter line

STATION FIX DEPTH: allows to fix depth to given value in corr.out, which can be useful if data is input to location program; set to 999. to disable

MAX DIFF TIME: maximum difference in seconds from average of correlated stations average, used to exclude relative times that are too large and could be due to cycle skipping.

MAX EVENT DISTANCE: maximum distance between event pair to compute correlation

MAX STAT DISTANCE: maximum distance between event and station

MIN CORR: minimum correlation required either for grouping or phase identification

MIN CORR CHAN: number of stations required for event pair to be correlated.

N DOUBLE SRATE: if sampling rate is to be increased give factor n to double sampling rate n times, 0. for none, this makes it possible to get phase reading with resolution greater than sampling interval

PRE SIGNAL: duration of signal to include before phase arrival if used

SINGLE BIT: the data can be reduced to 1 bit; reduce to 0 (-) and 1 (+) if set to 1, full data if set to 0.

START LATITUDE and START LONGITUDE: these can be set to write values to corr.out, which is in Nordic format and can be used as input for location programs; this can be used if all events analyzed belong to one cluster and the same starting location is to be used for all of them; set to 999. to disable

TRACE OUTPUT: the data can be written to corr.trace output file (1. for true)

WAVE CORR OUT: CORR can write out cross-correlation function and input traces to waveform output files of the selected duration, to disable set this parameter to 0., 1. for full data or 2. for reduced data, where 1 for data 

WAVENAME OUT: CORR writes out waveform filenames to corr.out, it is possible to either keep original waveform names (0.) or put corr output file names (1.), which after SPLIT of corr. out allows inspection of the results using eev and mulplt
Station parameters:

**STATION**: one line for configuration of each channel

- **STAT, COMP**: station and component codes
- **SELCRIT**: 1=P, 2=S, 4=full trace
- **DURATION**: signal duration in seconds if (selcrit=4) starting from either P or S
- **FLOW, FHIGH**: filter limits for bandpass filter, can be disabled by FILTER (see above)

Example of STATION line:

```plaintext
KEYWORD...STAT......COMP......SELCRIT...DURATION..FLOW......FHIGH.....
--- p ---
STATION PCA S Z 1. 6. 3. 8.
--- s ---
STATION EDI S E 2. 5. 3. 8.
```

Output files:

- **corr.out**: This is the main output file. The file is in Nordic format and contains the phase readings if run in phase detection mode and can be used with the SEISAN location programs directly. In continuous mode, the file can contain more than one phase reading per channel. In group identification mode the file contains the event list, cross-correlation matrix and suggested groups of similar events.
- **corr.trace**: This file gives details of program run and can provide information on cause of errors. 

Input file for hypodd giving relative phase times and correlation (see hypodd manual for details), e.g.

```
# 1 2 0.0
GMK -0.136 0.940 P
GMK -0.136 0.977 S
PCA -0.142 0.963 P
PCA -0.142 0.967 S
PCO -0.152 0.968 P
PCO -0.152 0.952 S
```

- **cc_pairs.out**: List of event pairs giving, index and s-file of first event, index and s-file of second event, number of stations and average correlation of all stations. This file is used as input to XCLUST.

```
 1 05-0028-32L.S200405 2 05-0413-55L.S200405 4 0.934
 1 05-0028-32L.S200405 3 05-1644-21L.S200405 2 0.838
 1 05-0028-32L.S200405 4 05-2102-06L.S200405 1 0.844
 1 05-0028-32L.S200405 5 05-2143-52L.S200405 3 0.905
 1 05-0028-32L.S200405 6 06-2136-12L.S200405 3 0.880
 1 05-0028-32L.S200405 7 06-2301-54L.S200405 4 0.901
```

**XCLUST**

XCLUST is a simple program for cluster analysis of output from program CORR (**cc_pairs.out**) to identify groups of similar events. This is done in a rather simple approach:
• sort event pairs with descending correlation

• find group
  o start with highest correlation
  o add events that are linked into group this group in several loops over all pairs until no more
    events can be added to group; link into group is given by one of the events in the pair being
    correlated with any of the events in the cluster

• continue to find next group

Visual inspection of the waveforms is highly recommended to confirm the clustering results.

Input file: xclust.par

This is the file for the main parameters, which are:
MINIMUM CORRELATION: minimum correlation required for pair to be used
MINIMUM STATIONS: minimum number of correlated stations required for pair to be used
MINIMUM PERGROUP: minimum number of events required to make a group
TRACE OUTPUT: flag to write trace output file (1. for true)

Output files:

xclust.trace: gives some details of what the program does, useful for debugging
xclust.out: gives list of events for each cluster and for each event the number of links with other events
  in that cluster

============
  group: 1 number of events: 20
-------------
  event links
-------------
   5   7
   11  8
    6   5
    2   7
    1   6
    7   9
...

Index.xxx: Index file where xxx refers to number of cluster. This file can be used with eev (e.g. eeve
  index.001) to work on a specific cluster.
Chapter 44

Programming in SEISAN and list of files in SEISAN distribution

This chapter gives a bit more technical details of SEISAN starting with a short programmers guide with description of sample and test programs.

44.1 Programmers guide and some test programs

SEISAN is conglomerate of programs and subroutines and it can be difficult to find out which routines to use and how to start a new SEISAN program. The most common method is to use an existing program and modify it. The intention with this section is to make it easier by providing a few sample programs which then can be modified to do specific tasks. The most common subroutines in LIB, which might be useful in other programs, are listed in Appendix D. The compilation of existing SEISAN programs has been described in section 3.10 and details of the commands are found in the Makefiles. In this distribution, sample programs have been included, which each illustrate the used of some SEISAN features. All programs are included in the Makefiles and can therefore be compiled directly, modified and recompiled.

Reading and writing S-files

A basic operation is to be able to read and write S-files, since all parameters are contained in the S-files. Starting with version 7.2, a new library (rea.for) and include block (rea.inc for definition of variables) has been included to make it easier to read and write data into S-files. Earlier, S-files were only read and written as text strings and individual parameters were then read/written to the text strings. Now the new routines do it all. These routines are now used in a few programs, but will be included whenever a program is substantially modified. The sample program is called sample_read_write.s.for. The program illustrates how to read all parameters in an S-file, make modifications and write out the file again. The program can be useful, if the user needs a program where special parameters are needed for a particular analysis or for output in another format. This program is written in Fortran.

The program can read from a file or a data base and therefore illustrates the different possibilities in SEISAN.

There is also a program available to read and write S-files in C. sample_read_write_s.c.cpp. The
program will use the seisan fortran routines to read and write, and the data is placed into a C structure. This C-struction is equivalent with the corresponding REA.INC common block in Fortran. In c++, the definitions are found in textttrea_c.h in INC but the definition of the variables are in rea.inc. All the c++ subroutines used are found in the main program. This program only read and write an S-file with one or many events.

Putting two S-files into memory at the same time, sample_rea_to_rea2.

Sometimes it is useful to have two s-files with all parameters in memory at the same time for different manipulations. Normally the rea.inc structure is used for one file. However, there is also a rea2.inc structure containing the same element as the rea structure, but all names are now rea2.... In rea2.for there are several subroutines that copy between rea and rea2 and the sample program illustrates this. The program operation is:

Data from 2 different s-files are read, one event at a time in each.
The two files are merged together, file 1 is inserted into file 2, so file 2 main header remains.
Id line from file 1 is converted to a comment line.
Locality line is only kept from file2, from file 1, it is converted to a comment.
If duplicate phases are found, they are deleted from file 1.

Reading and writing waveform files: SAMPLE_READ_WAV, SAMPLE_READ_CONT, SAMPLE_READ_ARC, SAMPLE_READ_WRITE_SEED, SAMPLE_WRITE_WAV

In SEISAN, waveform files can be in SEISAN, SAC, MiniSeed/SEED, Guralp or GSE format. SEISAN format is slightly different depending on which computer platform it is written and byte swapping has to be done in some cases. In order to automatically handle the reading of waveform files, irrespective of format and computer platform, a set of standard routines are used (waveform.for) and an include block where all parameters and data end up (waveform.inc). The sample reading program is called sample_read_wav.for. There is a similar program sample_write_wav for writing SEISAN waveform files.

The program illustrates how to read many waveform files belonging to one event as if it was one file, irrespective of format. It also demonstrates how to read just one waveform file. There is an output file which gives a listing of all different channels found in all the files read. This listing is in the format used for defining channels in an archive. There is no detail on how to write a SEISAN binary file in this program, but some info is given under the format description in Appendix B and the program tsig.for described below illustrates a simple write. The sample program sample_read_cont illustrates how to extract out a time segment of the continuous data base. The program also shows how to write a Seisan file with all headers. The program is started from the command prompt:
sample_read_cont start_time interval

where start time is yyyyymmddhhmmss and interval is interval in minutes.

There is a similar program for reading data from archives, sample_read_arc.
sample_read_write_seed interval

This program can read and write seed files using Chad and Rubens routines. It works independently of SEISAN subroutines.

SAMPLE_READ_WRITE_SEED There is also a routine in Java available to read all SEISAN binary formats. The program is called SFORMAT (written by T. Utheim). Similarly there is a sample program to read all SEISAN binary formats in Perl (written by Angel Rodriguez). The program is called seibinasc.pl and you need a Perl interpreter to run it. Before starting the program, a DIRF must first
be made of waveform input files. The output is identical to a SEISAN ASCII file as made by SEIASC. The sample write program is called `sample_write_wav.for`. It is a simple example of writing a straight line. The output form is SEISAN.

Correction for instrument response, sample_instrument_correction

This program will mainly demonstrate how to make instrument correction, but is also demonstrates several other standard subroutine calls in SEISAN. The operations are:

- Read S-file with readings and location
- Get waveform files from S-file
- Enter station and component for instrument correction
- Find S-time from readings
- Select out a time window for waveform channel around S-time
- Find channel number in waveform file(s) corresponding to desired channel
- Read the S-time window from waveform channel
- Read response file
- Prepare response removal, different filters and poles and zeros possible, the example is Wood-Anderson response
- Correct for instrument response
- Write out corrected data in an ASCII file in Helberger format
- Automatically pick maximum amplitude and corresponding period
- Write out results

Graphics in SEISAN

SEISAN uses a set of graphics routines, which are identical in call on all 3 platforms. These routines then call low level routines which are platform dependent (X on Unix and Windows calls on PC). The programmer only has to use the high level routines. The routines also generate a PostScript output if a given parameter is set. The program is called `sample_graphics.for`. The program illustrates how to initiate graphics, make a few simple calls, get up and use the mouse and make a hard copy file. Most of the general graphics routines are located in file `seiplot.for` and common variables in `seiplot.inc`. The program can be useful for testing functionality of the mouse.

Program LSQ is a simple example of how to make xy-graphics. It also shows how to make the output files for gmtxy. In order to find more info (apart from manual in INF) on gmtxy, see file `gmt.for` in LIB and `gmt_xxy.par` in INF.

Program for UDP communication SAMPLE_UDP_MESSAGE

This program shows how to communicate between two program using UDP signalling. It is so far only used in MULPLT to communicate with SE.

Program to make test signals, TSIG:

It is often useful to be able to work with controlled waveform data so a program making test signals is included. The program makes several traces, all with same length and sample rate and trace 1 is the sum of all traces. For each trace selected, the parameters selected are: Frequency, amplitude (remember this is integer numbers in file so use at least 1000), phase, delay (delay time when the signal appears
on trace relative to start of trace, the data before is zero) and damping. The damping is used to simulate seismometer damping or simple a damped signal and has a similar physical meaning as the seismometer damping constant, but period is not recalculated to simulate changing period with damping. Zero damping is no damping.

An additional trace can be made with a Brune displacement pulse generated with parameters corner frequency ($f_0$), $Q$ and kappa (see MULPLT) and travel time. Travel time is used for $Q$-correction and also places the pulse at travel time distance from the origin (start of trace), so length of trace must be longer than travel time. If zero $q$ and kappa, no attenuation is used. The program also write an S-file with relevant parameters. The program illustrates a simple writing of a SEISAN waveform file.

Java programs in SEISAN

The Java programs are each given as a Jar-file, the jar-files are located in the PRO directory. The jar-file contains all the Java source code, the Java classes and the project file so a user can decompress the jar file, change the script and make a new version of the program. The programs are started using a script file in the COM directory and no classpath has to be set, when SEISAN has been correctly installed.

44.2 Routines for Geodetic computations (WGS84)

The SEISAN software libaray include two subroutines for geodetic computations using the WGS84 ellipsoid:

- vincenty_forward($lon, lat, azi, dist$) and vincenty_inverse($lo, la, lo2, la2$)
- vincenty_forward($lon, lat, azi, dist$) computes the lat and lon for a point at an azimuth and distance (meter) from the point given by input lon and lat. Output lon and lat are returned at the position of input lon and lat.
- vincenty_inverse($lo, la, lo2, la2$) computes the azimuth, back-azimuth and distance between the two points $lo, la, lo2, la2$. Output azimuth, back-azimuth and distance (meters) are returned at the position of $lo, la, lo2, la2$.

The code is based on the code at [http://www.ngs.noaa.gov/PC_PROD/Inv_Fwd/](http://www.ngs.noaa.gov/PC_PROD/Inv_Fwd/) see also [http://www.movable-type.co.uk/scripts/latlong-vincenty.html](http://www.movable-type.co.uk/scripts/latlong-vincenty.html)

Please check for bugs before use.

Example:

```c
C program example that show how to call the Vincenty routines:

double precision lo, la, azi, dist, lo2, la2
real lon, lat
lon=12.8833
lat=56.54
azi=0.0D0
dist=110000.0D0 ! meters
lo=DBLE(lon)
lad=DBLE(lat)
lo2=DBLE(lon)
la2=DBLE(lat)
write(*,*)"lonlat:",lo,la
do i=1,3
write(*,*)"FORWARD:",lo,la,azi,dist,lo2,la2
```

Please check for bugs before use.

Example:
lo=DBLE(lon)
la=DBLE(lat)
write(*,*)"Input [lon,lat,azi, dist]: ",lo,la,azi,dist
write(*,*)"Input: lo1 la1 lo2 la2: ",lo,la,lo2,la2
call vincenty_inverse(lo,la,lo2,la2)
write(*,*)"Output: [azi,baz, dist]: ",lo,la,lo2
azi=azi+DBLE(90.0)
write(*,*)"-----------------------------------"
enddo
end
C End of example

44.3 CONTENTS OF PRO, LIB, INC, INF, COM, DAT, SUP, ISO and PIC DIRECTORIES

The PRO, LIB, INC and COM directories contain software, the DAT directory parameter files for operating the SEISAN system and INF contains documentation and manuals. All files are listed and explained in the file seisan.all in the INF directory.

The ISO directory contains macroseismic information.

The program CHECK can check if a distribution is complete. Run CHECK and use option ‘basic’. The content of the distribution is compared to the seisan.all file in the INF directory.

The seisan.all file also list programs no longer compiled, but with source code included in case there should be a future need for these programs.
Bibliography


BIBLIOGRAPHY


Appendix A

The Nordic format

Free columns are included for two purposes:

1. To obtain a readable format
2. To have some space for possible future extensions

Here are examples, top 3 lines for positioning only.

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1234567890123456789012345678901234567890123456789012345678901234567890123456789</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

-------------------------------------------------------------------------------

<table>
<thead>
<tr>
<th>1996</th>
<th>6</th>
<th>3</th>
<th>1955</th>
<th>35.5</th>
<th>D</th>
<th>47.760</th>
<th>153.227</th>
<th>0.0</th>
<th>TES</th>
<th>12</th>
<th>1.1</th>
<th>5.6WHRV</th>
<th>5.6bPDE1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1996</td>
<td>6</td>
<td>3</td>
<td>1955</td>
<td>35.5</td>
<td>D</td>
<td>47.760</td>
<td>153.227</td>
<td>0.0</td>
<td>TES</td>
<td>12</td>
<td>1.1</td>
<td>5.6WHRV</td>
<td>5.6bPDE1</td>
</tr>
<tr>
<td>GAP=348</td>
<td>2.88</td>
<td>999.9</td>
<td>999.9999.9</td>
<td>-0.1404E+08</td>
<td>-0.3810E+08</td>
<td>0.1205E+09</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1996 0603 1955 31.8 D 46.787153.722 33.0 PDE 5.6bPDE 1
ACTION:SPL 08-10-02 10:19 OP:jh STATUS: ID:19960603195540 1
1996-06-03-2002-18S.TEST__012 6
1996-06-03-1917-52S.TEST__.002 6

STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO AIN AR TRES W DIS CAZ7
KBS BZ EP 20 4 40.63 23 -1.3210 5724 351
TRO SZ EP 20 5 32.5 21 1.7510 6471 343
LOF SZ IP C 20 5 46.68 21 -0.1110 6729 344
JNW SZ EP 20 5 49.5 21 1.1910 6755 353
JMI LZ I 20 8 27.35 6768 353
JMI LZ I 2014 41.56 6768 353
JMI LZ I 2021 25.49 6768 353
MOL SZ IP C 20 6 25.49 19 -1.7410 7408 343
FOO SZ EP 20 6 35.99 19 0.1210 7559 344
HYA SZ EP 20 6 36.91 19 -0.1410 7580 343
SUE SZ IP C 20 6 39.07 19 -0.2810 7621 344
KONO BZ IP C 20 6 40.72 19 -0.7010 7657 341
ASK SZ EP 9 20 6 37.24 19 -4.94 0 7671 344
BER SZ EP 9 20 6 37.43 19 -5.16 0 7678 344

537
APPENDIX A. THE NORDIC FORMAT

<table>
<thead>
<tr>
<th>Location</th>
<th>Type</th>
<th>Number</th>
<th>Longitude</th>
<th>Latitude</th>
<th>Depth</th>
<th>Magnitude</th>
<th>Date</th>
<th>Time</th>
<th>Lat.</th>
<th>Long.</th>
<th>Depth</th>
<th>Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>EGD</td>
<td>SZ</td>
<td>9</td>
<td>20</td>
<td>6.5</td>
<td>38.42</td>
<td>17979.3100</td>
<td>19</td>
<td>-4.95</td>
<td>0</td>
<td>7692</td>
<td>344</td>
<td>1.731082</td>
</tr>
<tr>
<td>ODD1</td>
<td>SZ</td>
<td>20</td>
<td>6</td>
<td>6.5</td>
<td>45.57</td>
<td>217504.9100</td>
<td>19</td>
<td>1.731</td>
<td>0</td>
<td>7699</td>
<td>343</td>
<td>0.518010</td>
</tr>
<tr>
<td>BLS5</td>
<td>SZ</td>
<td>20</td>
<td>6</td>
<td>6.5</td>
<td>46.33</td>
<td>2175206.0000</td>
<td>19</td>
<td>-0.50</td>
<td>0</td>
<td>7753</td>
<td>343</td>
<td>-0.50106</td>
</tr>
</tbody>
</table>

Below are examples of how the last free columns of type 4 lines are used in the Nordic Databank in Helsinki and in Bergen:

```
1985 510 21 5 16.1 LE 60.240 6.170 30.0F BER 6 2.3 3.8LNAO 4.0bPDE 3.2sISC 1
   1.5 0.5 0.9 5.0 0.4 5
8505210425.WNN                               6
ACTION: UPD 93-07-09 09:40 OP:jens STATUS: ID:19920101080359 I
STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO AIN AR TRES W DIS CAZ7
STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO AIN AR TRES W DIS CAZ7
8505210425.WNN                               6
ACTION: UPD 93-07-09 09:40 OP:jens STATUS: ID:19920101080359 I
STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO AIN AR TRES W DIS CAZ7
NRSA SZ IPN 1 D 2105 13.44 0345 1234.6 1.33 245.2 08.6 22 2 -0.7 9 555 235
BER SZ IPG 2 U 2105 25.41 200
HYA SZ ISG 1 2105 33.1
ODD SZ IP 3 2105 20.1 250
ODD SZ EPG 2105 22.9
ODD SZ LG 2105 55.8
```

Note in this example the fault plane solution line(F) and the HYP error line(E).

```
1993 1028 0800 26.4 L 57.518 7.119 18.8 BER 6 .6 2.6CBER 1
   1.20 6.4 7.0 6.8 .3359E+01 -.2719E+00 .3054E+02E
93.2 74.8 -48.2 2 F
ACTION: SPL 95-01-08 09:40 OP:jh STATUS: ID:19931028080019 I
9310-28-0800-19S.NSN__17 6
STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO AIN AR TRES W DIS CAZ7
BLS5 SZ EP D8 0 56.80 129 -.110 216 349
BLS5 SZ ESG 8 1 23.59 -.910 216 349
BLS5 SZ EP 8 0 56.80 129 -.110 216 349
BLS5 SZ ESG 8 1 23.59 -.910 216 349
```

Location parameters:

AR: Azimuth residual when using azimuth information in locations
TRES: Travel time residual
W: Actual weight used for location (inc. e.g. distance weight), i2
DIS: Epicentral distance in km
CAZ: Azimuth from event to station

---

Note: Type 1 line must be the first, all type 4 lines should be together and the last line must be blank
FORMAT DESCRIPTION:

Type 1 Line: Hypocenter line

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Free</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-5</td>
<td>I4</td>
<td>Year</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Free</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7-8</td>
<td>I2</td>
<td>Month</td>
<td></td>
</tr>
<tr>
<td>9-10</td>
<td>I2</td>
<td>Day of Month</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>Fix o. time</td>
<td>Normally blank, an F fixes origin time</td>
</tr>
<tr>
<td>12-13</td>
<td>I2</td>
<td>Hour</td>
<td></td>
</tr>
<tr>
<td>14-15</td>
<td>I2</td>
<td>Minutes</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Free</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17-20</td>
<td>F4.1</td>
<td>Seconds</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>A1</td>
<td>Location model indicator</td>
<td>Any character</td>
</tr>
<tr>
<td>22</td>
<td>A1</td>
<td>Distance Indicator</td>
<td>L = Local, R = Regional, D = Distant, etc.</td>
</tr>
<tr>
<td>23</td>
<td>A1</td>
<td>Event ID</td>
<td>E = Confirmed explosion, P = Probable explosion, V = Volcanic, Q = Confirmed earthquake, ' ' = Presumed earthquake, L = Landslide (was X), S = Acoustic signal, I = triggered or induced, O = confirmed other event, C = Cryogenic event (e.g. ice quake), G = Glacial earthquake (subset of C)</td>
</tr>
<tr>
<td>24-30</td>
<td>F7.3</td>
<td>Latitude</td>
<td>Degrees (+ N)</td>
</tr>
<tr>
<td>31-38</td>
<td>F8.3</td>
<td>Longitude</td>
<td>Degrees (+ E)</td>
</tr>
<tr>
<td>39-43</td>
<td>F5.1</td>
<td>Depth</td>
<td>Km</td>
</tr>
<tr>
<td>44</td>
<td>A1</td>
<td>Depth Indicator</td>
<td>F = Fixed, S = Starting value</td>
</tr>
<tr>
<td>45</td>
<td>A1</td>
<td>Locating indicator</td>
<td>----------------------------, * do not locate</td>
</tr>
<tr>
<td>46-48</td>
<td>A3</td>
<td>Hypocenter Reporting Agency</td>
<td></td>
</tr>
<tr>
<td>49-51</td>
<td></td>
<td>Number of Stations Used</td>
<td></td>
</tr>
<tr>
<td>52-55</td>
<td></td>
<td>RMS of Time Residuals</td>
<td></td>
</tr>
<tr>
<td>56-59</td>
<td>F4.1</td>
<td>Magnitude No. 1</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>A1</td>
<td>Type of Magnitude L=ML, b=mb, B=mB, s=Ms, s=MS, W=MW, G=MbLg (not used by SEISAN), C=Mc</td>
<td></td>
</tr>
<tr>
<td>61-63</td>
<td>A3</td>
<td>Magnitude Reporting Agency</td>
<td></td>
</tr>
<tr>
<td>64-67</td>
<td>F4.1</td>
<td>Magnitude No. 2</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>A1</td>
<td>Type of Magnitude</td>
<td></td>
</tr>
<tr>
<td>69-71</td>
<td>A3</td>
<td>Magnitude Reporting Agency</td>
<td></td>
</tr>
<tr>
<td>72-75</td>
<td>F4.1</td>
<td>Magnitude No. 3</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>A1</td>
<td>Type of Magnitude</td>
<td></td>
</tr>
<tr>
<td>77-79</td>
<td>A3</td>
<td>Magnitude Reporting Agency</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>A1</td>
<td>Type of this line (&quot;1&quot;), can be blank if first</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX A. THE NORDIC FORMAT

line of event

If more than 3 magnitudes need to be associated with the hypocenter in the first line, a
subsequent additional type one line can be written with the same year, month, day until event
ID and hypocenter agency. The magnitudes on this line will then be associated with the main
header line and there is then room for 6 magnitudes.

Type 2 line (Macroseismic information)

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>Blank</td>
</tr>
<tr>
<td>6-20</td>
<td>a Any descriptive text</td>
</tr>
<tr>
<td>21</td>
<td>Free</td>
</tr>
<tr>
<td>22</td>
<td>a1 Diastrophism code (PDE type)</td>
</tr>
<tr>
<td></td>
<td>F = Surface faulting</td>
</tr>
<tr>
<td></td>
<td>U = Uplift or subsidence</td>
</tr>
<tr>
<td></td>
<td>D = Faulting and Uplift/Subsidence</td>
</tr>
<tr>
<td>23</td>
<td>a1 Tsunami code (PDE type)</td>
</tr>
<tr>
<td></td>
<td>T = Tsunami generated</td>
</tr>
<tr>
<td></td>
<td>Q = Possible tsunami</td>
</tr>
<tr>
<td>24</td>
<td>a1 Seiche code (PDE type)</td>
</tr>
<tr>
<td></td>
<td>S = Seiche</td>
</tr>
<tr>
<td></td>
<td>Q = Possible seiche</td>
</tr>
<tr>
<td>25</td>
<td>a1 Cultural effects (PDE type)</td>
</tr>
<tr>
<td></td>
<td>C = Casualties reported</td>
</tr>
<tr>
<td></td>
<td>D = Damage reported</td>
</tr>
<tr>
<td></td>
<td>F = Earthquake was felt</td>
</tr>
<tr>
<td></td>
<td>H = Earthquake was heard</td>
</tr>
<tr>
<td>26</td>
<td>a1 Unusual events (PDE type)</td>
</tr>
<tr>
<td></td>
<td>L = Liquefaction</td>
</tr>
<tr>
<td></td>
<td>G = Geysir/fumerol</td>
</tr>
<tr>
<td></td>
<td>S = Landslides/Avalanches</td>
</tr>
<tr>
<td></td>
<td>B = Sand blows</td>
</tr>
<tr>
<td></td>
<td>C = Cracking in the ground (not normal faulting).</td>
</tr>
<tr>
<td></td>
<td>V = Visual phenomena</td>
</tr>
<tr>
<td></td>
<td>O = Olfactory phenomena</td>
</tr>
<tr>
<td></td>
<td>M = More than one of the above observed.</td>
</tr>
<tr>
<td>27</td>
<td>Free</td>
</tr>
<tr>
<td>28-29</td>
<td>i2 Max Intensity</td>
</tr>
<tr>
<td>30</td>
<td>a1 Max Intensity qualifier (+ or - indicating</td>
</tr>
<tr>
<td></td>
<td>more precicely the intensity)</td>
</tr>
<tr>
<td>31-32</td>
<td>a2 Intensity scale (ISC type defintions)</td>
</tr>
<tr>
<td></td>
<td>MM = Modified Mercalli</td>
</tr>
<tr>
<td></td>
<td>RF = Rossi Forel</td>
</tr>
<tr>
<td></td>
<td>CS = Mercalli - Cancani - Seberg</td>
</tr>
<tr>
<td></td>
<td>SK = Medevev - Sponheur - Karnik33 Free</td>
</tr>
<tr>
<td>34-39</td>
<td>f6.2 Macroseismic latitude (Decimal)</td>
</tr>
<tr>
<td>40</td>
<td>Free</td>
</tr>
<tr>
<td>41-47</td>
<td>f7.2 Macroseismic longitude (Decimal)</td>
</tr>
<tr>
<td>48</td>
<td>Free</td>
</tr>
</tbody>
</table>
Macroseismic magnitude

- Magnitude based on maximum Intensity.
  - A = Magnitude based on felt area.
  - R = Magnitude based on radius of felt area.
  - * = Magnitude calculated by use of special formulas developed by some person for a certain area.

  Further info should be given on line 3.

Logarithm (base 10) of radius of felt area.

Logarithm (base 10) of area (km**2) number 1 where earthquake was felt exceeding a given intensity.

Intensity bordering the area number 1.

Logarithm (base 10) of area (km**2) number 2 where earthquake was felt exceeding a given intensity.

Intensity bordering the area number 2.

Quality rank of the report (A, B, C, D)

Reporting agency

Type 3 Line (Optional): Comments, more examples below

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Free</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-79</td>
<td>A</td>
<td>A Text</td>
<td>Anything</td>
</tr>
<tr>
<td>80</td>
<td>A1</td>
<td>Type of this line (&quot;3&quot;)</td>
<td></td>
</tr>
</tbody>
</table>

This type of line can be used to specify xnear, xfar and the starting depth for use with HYPOCENTER. For example

XNEAR 200.0 XFAR 400.0 SDEP 15.0 3

8-13 f6.1 Xnear
20-25 f6.1 Xfar
32-36 f5.1 Starting depth

Type 4 line: Phase lines

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Free</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2-6</td>
<td>A5</td>
<td>Station Name Blank = End of readings = end of event</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>A1</td>
<td>Instrument Type S = SP, I = IP, L = LP etc</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>A1</td>
<td>Component Z, N, E, T, R, 1, 2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Free</td>
<td>or weight, see note below</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>A1</td>
<td>Quality Indicator I, E, etc.</td>
<td></td>
</tr>
<tr>
<td>11-14</td>
<td>A2</td>
<td>Phase ID PN, PG, LG, P, S, etc. **</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>I1</td>
<td>Weighting Indicator (1-4) 0 or blank= full weight, 1=75%, 2=50%, 3=25%, 4=0%, 9: no weight</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Free</td>
<td>or flag A to indicate automartic pick, removed when picking</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX A. THE NORDIC FORMAT

17 A1 First Motion  C, D
18 Note: Currently 15 to 18 can also be used for phase assuming
column 11-14 is not blank. See note ** below.
19-20 I2 Hour  Hour can be up to 48 to
  indicate next day
21-22 I2 Minutes
23-28 F6.0 Seconds
29 Free
30-33 I4 Duration (to noise)  Seconds
34-40 g7.1 Amplitude (Zero-Peak) in units of nm, nm/s, nm/s^2 or counts.
41 Free
42-45 F4.0 Period  Seconds
46 Free
47-51 F5.0 Direction of Approach, back azimuth  Degrees
52 Free
53-56 F4.0 Phase Velocity  Km/second
57-60 F4.0 Angle of incidence (was Signal to noise ratio before version 8.0)
61-63 I3 Back azimuth residual
64-68 F5.1 Travel time residual
69-70 I2 Weight
71-75 F5.0 Epicentral distance(km)
76 Free
77-79 I3 Azimuth at source
79 A1 Type of this line ("4"), can be blank, which it is
  most often

NB: Epicentral distance: Had format I5 before version 7.2. All old lines can
be read with format F5.0 with same results, but now distance can also be
e.g. 1.23 km which cannot be read by earlier versions. However, an UPDATE
would fix that.

** Long phase names: An 8 character phase can be used in column 11-18. There
is then not room for polarity information. The weight is then put into column 9.
This format is recognized by HYP and MULPLT.

Type 4 cards should be followed by a Blank Card (Type 0)

Type 5 line (optional): Error estimates of previous line, currently not used
by any SEISAN programs.

Columns Format Description  Comments
  1 Free
  2-79 Error estimates in same format as previous line, normally type 4
  80 A1 Type of this line ("5")

Type 6 Line (Optional): Reference to where waveform data is found.
This is mostly a file name but can also be a reference to a position
in an archive.
Columns Format Description Comments
1 Free
2-79 Name of file or archive reference, a-format,
80 Type of this line ("6")

Archive reference:
Example:

```
ARC STAT COM NT LO YYYYY MMDD HHMM SS DUR
ARC ROSA BHZ PM 2010 1011 0100 00 14400
```

1 Free
2-04 Must be the keyword ARC
5 Free
6-10 Station (a5)
11 Free
12-14 Component (a3)
15 Free
16-17 Network (a2)
18 Free
19-20 Location (a2)
21 Free
22-38 Year, month, day, hour, min, sec (i4,1x,2i2,1x,2i2,1x,i2)
39 Free
40-44 Duration of extract (s), (i5)
46-79 Free
80 Type of this line ("6")

Type 7 Line (Optional): Title line for phases, fixed text

Columns Format Description Comments
1 Free
2-79 A Help lines to place the numbers in right positions
80 A1 Type of this line ("7")

Example

```
STAT SP IPHASW D HRMM SECON CODA AMPLIT PERI AZIMU VELO AIN AR TRES W DIS CAZ7
```

Type E Line (Optional): Hyp error estimatea of prime hypocenter, only one
E-lie in S-file

Columns Format Description
APPENDIX A. THE NORDIC FORMAT

1 Free
2 - 5 A4 The text GAP=
6 - 8 I3 Gap
15-20 F6.2 Origin time error
25-30 F6.1 Latitude (y) error
31-32 Free
33-38 F6.1 Longitude (x) error (km)
39-43 F5.1 Depth (z) error (km)
44-55 E12.4 Covariance (x,y) km*km
56-67 E12.4 Covariance (x,z) km*km
68-79 E14.4 Covariance (y,z) km*km

Type F Line (Optional): Fault plane solution

Columns Format Description

1:30 3F10.0 Strike, dip and rake, Aki convention
31:45 3F5.1 Error in strike dip and rake (HASH), error in fault plane and aux. plane (FPFIT)
46:50 F5.1 Fit error: FPFIT and HASH (F-fit)
51:55 F5.1 Station distribution ratio (FPFIT, HASH)
56:60 F5.1 Amplitude ratio fit (HASH, FOCMEC)
61:62 I2 Number of bad polarities (FOCMEC, PINV)
64.65 I2 Number of bad amplitude ratios (FOCMEC)
67:69 A3 Agency code
71:77 A7 Program used
78:78 A1 Quality of solution, A (best), B C or D (worst), added manually
79:79 A1 Blank, can be used by user
80:80 A1 F

Type H line, High accuracy hypoenter line

Columns

1:55 As type 1 line
16 Free
17 Seconds, f6.3
23 Free
24:32 Latitude, f9.5
33 Free
34:43 Longitude, f10.5
44 Free
45:52 Depth, f8.3
53 Free
54:59 RMS, f6.3
60:79 Free
80 H

Type I Line, ID line

Columns Format description1 Free

  2:8 Help text for the action indicator
  9:11 Last action done, so far defined

    ARG: AUTO Register, AUTOREG
    ARX: From ARC extract
    DPH: Phases deleted with DELS
    DUP: Duplicated event
    HIN: Updated with HYPOINVERSE
    NEW: New event
    REE: Register from EEV
    REG: Register
    SPL: Split
    UPD: Update with HYP
    UP: Update with HYP only from EEV

12 Free
13:26 Date and time of last action
27 Free

28:30 Help text for operator

36:42 Help text for status

43:56 Status flags, not yet defined

57 Free

58:60 Help text for ID

61:74 ID, year to second

75 If d, this indicate that a new file id had to be created which was
  one or more seconds different from an existing ID to avoid overwrite.

76 Indicate if ID has been synchronized with origin time.

  Blank means not synchronized, S means synchronized. In version 11 and before
  this was L. Only for information.

Type M Line (Optional): Moment tensor solution

Note: the type M lines are pairs of lines with one line that gives the hypocenter
  time, and one line that gives the moment tensor values:

The first moment tensor line:

Columns Format Description

  1:1 Free
  2: 5 I4 Year
  7: 8 I2 Month
  9:10 I2 Day of Month
12:13 I2 Hour
14:15 I2 Minutes
17:20 F4.1 Seconds
APPENDIX A. THE NORDIC FORMAT

24:30 F7.3 Latitude Degrees (+ N)
31:38 F8.3 Longitude Degrees (+ E)
39:43 F5.1 Depth Km
46:48 A3 Reporting Agency
56:59 F4.1 Magnitude
60 A1 Type of Magnitude L=ML, b=mb, s=Ms, S=MS, W=MW,
61:63 A3 Magnitude Reporting Agency
71:77 A7 Method used
78:78 A1 Quality of solution, A (best), B C or D (worst), added manually
79:79 A1 Blank, can be used by user
80:A1 M

The second moment tensor line:

Columns Format Description
1:1 Free
2:3 A2 MT
4:9 F6.3 Mrr or Mzz [Nm] divided by exponential
11:16 F6.3 Mtt or Mxx [Nm] -------
18:23 F6.3 Mpp or Myy [Nm] -------
25:30 F6.3 Mrt or Mzx [Nm] -------
32:37 F6.3 Mrp or Mzy [Nm] -------
39:44 F6.3 Mtp or Mxy [Nm] -------
46:48 A3 Reporting Agency
49:49 A1 MT coordinate system (S=spherical, C=Cartesian)
50:51 i2 Exponential
53:62 G6.3 Scalar Moment [Nm]
71:77 A7 Method used
78:78 A1 Quality of solution, A (best), B C or D (worst), added manually
79:79 A1 Blank, can be used by user
80:80 A1 M

All moment values must be multiplied by the exponent

Type P line, file name of a picture file

1:1 Free
2:79 File name
80:80 P

Type E13 and EC3 line, explosion information

Example

1980 0124 0927 CHARGE(T): 0.5 E13 LE Haakonsvern, HAA underwater explosion E13 EC3

Information on explosion site, time and agency, same format as a type 1 line, no magnitudes used, last 3 columns must be E13
EC3 Information on charge and site
Columns
   2:11 Info text
   11:12 Blank
   13:22 Charge in tons, f10.3
   23:77 Any information, a
   78:80 EC3

Type MACRO3 line: File name of macroseismic observations in ISO directory

Example:

1980-03-14-0456-05.MACRO MACRO3

An example of the file is:

Sunnfjord 1980 314 456 5 GMT 1980 314 556 5 Local time
   Comment
   60.500 5.270 1.0 EMS 5088 MJÖELKERAAEN
   60.560 5.260 1.0 EMS 5100 ISDALSTOE
   60.570 5.050 1.0 EMS 5112 ROSSLAND

1. Line
   Location, GMT time, Local time. Format a30,i4,ix,2i2,ix,2i2,ix,2i2,ix,i2,'GMT',ix,i4,ix,2i2,ix,2i2,ix,2i2,ix,i2,ix,'Local time'
   2. Line Comments
   3. Line Observations: Latitude, Longitude, intensity, code for scale, postal code or similar, location. Format 2f10.4,f5.1,ix,a3,ix,a10,2x,a.
      Note the postal code is an ascii string and left justified (a10).

Type 3 line giving xnear/xfar

Definition of xnear and xfar to be used with HYPOCENTER.

Example
   XNEAR 1000.0 XFAR 2000.0 3
   Columns
   8-13: xnear value
   20-25: xfar value

Type 3 line with locality, example

   LOCALITY: atlantic ocean

Type 3 line with felt information, example

   FELTINFO: cracks in the ground at xx

New Nordic Format

So far the new format only affects type 4 lines. SPEC lines will also be changed but has not been done
APPENDIX A. THE NORDIC FORMAT

so far. Description below:

<table>
<thead>
<tr>
<th>Type 4 line: Phase lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns Format Description Comments</td>
</tr>
<tr>
<td>1 Free</td>
</tr>
<tr>
<td>2-6  A5 Station Name</td>
</tr>
<tr>
<td>7-9  A3 Component, eg SHZ</td>
</tr>
<tr>
<td>10 Free</td>
</tr>
<tr>
<td>11-12 A2 Network code</td>
</tr>
<tr>
<td>13-14 A2 Location</td>
</tr>
<tr>
<td>15 Free</td>
</tr>
<tr>
<td>16 A1 Quality Indicator I, E, etc.</td>
</tr>
<tr>
<td>17-24 A8 Phase ID PN, PG, LG, P, S, etc.</td>
</tr>
<tr>
<td>25 I1 Weighting Indicator (1-4) 0 or blank= full weight, 1=75%, 2=50%, 3=25%, 4=0%, 9: no weight, time (e.g. P-S).</td>
</tr>
<tr>
<td>26 Free or flag A to indicate automatic pick, removed when picking</td>
</tr>
<tr>
<td>27-28 I2 Hour Hour can be up to 48 to indicate next day</td>
</tr>
<tr>
<td>29-30 I2 Minutes</td>
</tr>
<tr>
<td>31 Free</td>
</tr>
<tr>
<td>32-37 F6.0 Seconds</td>
</tr>
<tr>
<td>38-44 Parameter 1. If a phase, it is polarity in column 44</td>
</tr>
<tr>
<td>If phase END for coda, it is duration(s) (F)</td>
</tr>
<tr>
<td>If amplitude phase, it is amplitude (G7)</td>
</tr>
<tr>
<td>Amplitude is (Zero-Peak) in units of nm, nm/s, nm/s~2 or counts.</td>
</tr>
<tr>
<td>If back azimuth phase (BAZ), it a back azimuth (F)</td>
</tr>
<tr>
<td>45-50 Parameter 2 If a phase or END, it is blank</td>
</tr>
<tr>
<td>If an amplitude, it is period (F)</td>
</tr>
<tr>
<td>If back azimuth phase, it is apparent velocity(km/s)</td>
</tr>
<tr>
<td>51 Free</td>
</tr>
<tr>
<td>52-54 A3 Agency</td>
</tr>
<tr>
<td>55 Free</td>
</tr>
<tr>
<td>56-58 A3 Operator</td>
</tr>
<tr>
<td>59 Free</td>
</tr>
<tr>
<td>60-63 F4.0 Angle of incidence</td>
</tr>
<tr>
<td>64-68 F5.1 Residual, can be travel time, back azimuth or magnitude</td>
</tr>
<tr>
<td>69-70 I2 Weight used, like 05 is 0.5</td>
</tr>
<tr>
<td>71-75 F5.0 Epicentral distance(km)</td>
</tr>
<tr>
<td>76 Free</td>
</tr>
<tr>
<td>77-79 I3 Azimuth at source</td>
</tr>
<tr>
<td>80 A1 Type of this line (&quot;4&quot;), can be blank, which it is most often</td>
</tr>
</tbody>
</table>

Example:

1996 6 7 1325 29.2 L 59.846 5.130 12.0F TES 12 .60 1.9LTES 2.2CTES 2.0LNA01
GAP=177 2.78 4.5 12.80.000 0.2239E+02 0.6258E+03 -0.2817E+03
<table>
<thead>
<tr>
<th>Year</th>
<th>Month</th>
<th>Day</th>
<th>Hour</th>
<th>Minute</th>
<th>Second</th>
<th>PAR1</th>
<th>PAR2</th>
<th>AGA</th>
<th>OPE</th>
<th>AIN</th>
<th>RES</th>
<th>W</th>
<th>DIS</th>
<th>CAZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1996</td>
<td>6</td>
<td>7</td>
<td>1325</td>
<td>30.5</td>
<td>59.76</td>
<td>5.396</td>
<td>29.2</td>
<td>NAO</td>
<td>2</td>
<td>1.0</td>
<td>2.0</td>
<td>L</td>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Coda is now phase end
Appendix B

The SEISAN waveform file format

The file is written from Fortran as an unformatted file. This means that the file contains additional characters (not described below, see end of this Appendix) between each block, which must be taken into account if the file is read as a binary file. If read as Fortran unformatted, the content will appear as described below. However, the internal structure is different on Sun, Linux and PC. SEISAN automatically corrects for these differences. The SEISAN ASCII format has identical headers to the binary files, however the binary samples are written as formatted integers, one channel at the time just like the in the binary format.

EVENT FILE HEADER

CONTAINS MINIMUM 12 ASCII STRINGS OF 80 BYTES. ALL FORMATS I OR A UNLESS OTHERWISE SPECIFIED.

line 1
1  1: FREE
 2 30: NETWORK NAME, COULD E.G. BE WESTERN NORWAY NETWORK
31 33: NUMBER OF CHANNELS, MAX 999
34 36: YEAR-1900, e.g. 101 for 2001 (I3)
37
38 40: DOY
41
42 43: MONTH
44
45 46: DAY
47
48 49: HR
50
51 52: MIN
53
54 59: SEC, FORMAT F6.3
60
61 69: TOTAL TIME WINDOW (SECS), FORMAT F9.3
APPENDIX B. THE SEISAN WAVEFORM FILE FORMAT

70 80: FREE
71
72
73 80: FREE

line 2
1 80: FREE

line 3
1
2 5: STATION CODE (A4), first 4 characters
6 7: FIRST two COMPONENT CODES (A2), SEED style
8 : NOT USED
9 : LAST COMPONENT CODE (A1), SEED style
10 : STATION CODE (A1), LAST CHARACTER IF 5 CHARACTER STATION CODE
11 17: START TIME RELATIVE TO EVENT FILE TIME (SECS) F7.2
18 : BLANK
19 26: STATION DATA INTERVAL LENGTH (SECS) F8.2
27 52: SECOND CHANNEL
53 78: THIRD CHANNEL
79 80: BLANK

line 4-XX, where XX depends on number of channels, however, XX
is at least 12 so there might be some blank lines.
1 80: THREE MORE CHANNELS (SAME FORMAT AS line 3)

EVENT FILE CHANNEL HEADER

HEADER IS 1040 BYTES LONG, WRITTEN AS ONE VARIABLE DEFINED AS
CHARACTER*1040
THE PARAMETERS ARE WRITTEN FORMATTED WITH INTERNAL WRITE INTO
1040 BYTE TEXT STRING.
FORMAT IS ALWAYS I FORMAT UNLESS OTHERWISE SPECIFIED

1 5: STATION CODE (A5)
6 7: FIRST TWO COMPONENT CODES (A2), SEED style
8 : FIRST LOCATION CODE (A1), SEED style
9 : LAST COMPONENT CODE (A1), SEED style
10 12: YEAR - 1900, e.g. 101 for 2001, (I3)
13 : SECOND LOCATION CODE (A1), SEED style
14 16: DOY
17 : FIRST NETWORK CODE (A1), SEED style
18 19: MONTH
20 19: SECOND NETWORK CODE (A1), SEED style
21 22: DAY
23  25: HR
26
27  28: MIN
29 : TIMING INDICATOR, BLANK: TIME IS OK, E: UNCERTAIN TIME
30  35: SECOND (F6.3)
36
37  43: SAMPLE RATE (F7.2 or any f-format)
38  50: NUMBER OF SAMPLES (I7)
39
40  59: LATITUDE (F8.4), optional
41
42  69: LONGITUDE (F9.4), optional
43
44  75: ELEVATION (METERS), optional
45  : Indicate gain factor: Blank: No gain factor, G: Gain factor in
46   column 148 to 159
47  : 2 OR 4 FOR 2 OR 4 BYTE INTEGER, BLANK IS 2 BYTE
48  : P: Poles and zeros used for response info, blank: Seismometer
49   period etc used for response info. See below for details.
50   T: Use up to 30 tabulated values irrespective of what is given
51   below. If less than 30, blank characters must be given.
52   C: a combination of table, poles and zeros or instrument
53   constants have been used, for information only. Value in 78
54   must then be T.
55   F: Force use of header response, e.g. generated by MULPLT. Only
56   gain at 1 hz is correct and 78 must be set to T.
57
58  80 - 80: FREE
59
60  148- 159: Normally comment, if 76 set to G, this is a gain factor,
61   format G12.7. All samples read from channel are multiplied
62   by this factor when read by routine seisinc. Used when data
63   is stored in units of e.g. nm where values can be less than 1.
64   Currently generated by MULPLT when option OUT is used to
65   extract part of a waveform file. Also program WAVETOOL will
66   generate these files (used with Out option). Some conversion
67   programs may also write this.
68
69  81 - 160: COMMENT LINE DESCRIBING THE SYSTEM RESPONSE (A80)
70
71  If character 78 is blank, option 1:
72
73  161 - 240: (10G8.3)  1. SEISMOMETER PERIOD
74    2. FRACTION OF CRITICAL DAMPING
75    3. SEISMOMETER GENERATOR CONSTANT (V/m/s) or
76      ACCELEROMETER SENSITIVITY (V/G)
77    4. AMPLIFIER GAIN
78    5. RECORDING MEDIA GAIN (I.E. 2048 COUNTS/VOLT)
79    6. GAIN AT 1.0 HZ, UNITS: COUNTS/METER
80    7. CUTOFF FREQUENCY FOR FILTER1 (HZ)
81    8. # OF POLES FOR FILTER1 (NEGATIVE FOR HIGHPASS)
APPENDIX B. THE SEISAN WAVEFORM FILE FORMAT

9. CUTOFF FREQUENCY FOR FILTER2 (HZ)

10. # OF POLES FOR FILTER2 (NEGATIVE FOR HIGHPASS)

241 - 320: (10G8.3) FREQUENCIES AND #’S OF POLES FOR FIVE MORE FILTERS

321 -1040: RESPONSE CURVES (9(10G8.3) FREQ., AMPL. (REL. 1.0 HZ) AND PHASE,
WRITTEN IN GROUPS OF 10 FREQUENCIES, 10 AMPLITUDES AND 10 PHASES

If character 78 is P, option 2:

161 - 182 (1X,2I5,6G11.4) 1. NUMBER OF POLES
2. NUMBER OF ZEROS
3. NORMALIZATION CONSTANT, COUNTS/M

183 - 240 (5G11.4) 2 Poles in pairs of real and imaginary parts
241 -1040 (G11.4) Remaining poles and zeros. 7 values are written
and then 3 spaces are left blank, see example below.

For each pole or zero, there are two real numbers representing the real and imaginary part of the pole
or zero, thus the number of poles is half the number of values written. First all the poles are written in
pairs of real and imaginary parts, then follow the zeros. There is room for a total of 37 poles and zeros
(74 pairs). The poles and zeros are written in a simulated line mode to make it easier to read, thus the 3
blanks after writing 7 values. It is assumed that the response is in displacement with units of counts/m.

SLR L E 86 199 7 18 15 6 35.960 1.000 1320 4P

| 11 | 5 | .2760E+11 | .3770 | .1830 | .3770 | .1830 | .6540 |
| .0000 | .2320 | .0000 | .2320 | .0000 | .2320 | .0000 |
| .3280 | .0000 | .3280 | .0000 | .3280 | .0000 | .2140E 01 |
| .0000 | .2140E 01 | .0000 | .0000 | .0000 | .0000 | .0000 |
| .0000 | .0000 | .0000 | .0000 | .0000 | .0000 |

NOTE: The component information in character 6 IS VERY IMPORTANT. It MUST be A if an ac-
ccelerometer is used, any other character assumes a velocity transducer. This is only relevant however
if option 1 is used where response values will be calculated from the free period etc. If option 1 with
discrete values or poles and zeros are used, the first component character can be anything.
To write a SEISAN file: If main headers are called mhead, channel header chead, data is data (integer), there is nchan channels and each has nsamp samples, then the file is written as

Do i=1,12
  Write(1) mhead(i)
Enddo
Do k=1,nchan
  Write(1) chead
  Write(1) (data(i),i=1,nsmap)
Enddo

This example only works up to 30 channels when writing main header. For more channels, see e.g. program SEISEI how to do it.

Details of binary file structure

When Fortran writes a files opened with "form=unformatted", additional data is added to the file to serve as record separators which have to be taken into account if the file is read from a C-program or if read binary from a Fortran program. Unfortunately, the number of and meaning of these additional
characters are compiler dependent. On Sun, Linux, MacOSX and PC from version 7.0 (using Digital Fortran), every write is preceded and terminated with 4 additional bytes giving the number of bytes in the write. On the PC, Seisan version 6.0 and earlier using Microsoft Fortran, the first 2 bytes in the file are the ASCII character "KP". Every write is preceded and terminated with one byte giving the number of bytes in the write. If the write contains more than 128 bytes, it is blocked in records of 128 bytes, each with the start and end byte which in this case is the number 128. Each record is thus 130 bytes long. All of these additional bytes are transparent to the user if the file is read as an unformatted file. However, since the structure is different on Sun, Linux, MacOSX and PC, a file written as unformatted on Sun, Linux or MacOSX cannot be read as unformatted on PC or vice versa. The files are very easy to write and read on the same computer but difficult to read if written on a different computer. To further complicate matters, the byte order is different on Sun and PC. With 64 bit systems, 8 bytes is used to define number of bytes written. This type of file can also be read with SEISAN, but so far only data written on Linux have been tested for reading on all systems. This means that version 7.0 can read all earlier waveform files on all platforms from all platforms. However, files written on version 7.0 PC cannot be read by any earlier versions of Seisan without modifying the earlier seisan version. In SEISAN, all files are written as unformatted files. In order to read the files independently of where they were written, the reading routine (buf_read in seisinc, in LIB) reads the file from Fortran as a direct access file with a record length of 2048 bytes. The additional bytes are thrown away, the relevant bytes fished out and swapped if the file is written on a different computer than where it is read. Since there is no information stored in the header of the file giving the byte address of each channel, the routine must read the first file-header, calculate how many bytes there are down to where the next channel starts, jump down and repeat the process until the desired channel is reached (this is also how SUDS files are read). However, compared to reading the file as unformatted, only a fraction of the file is read to fish out a particular channel. Once the channel header has been read, the start address is stored in the subroutine so any subsequent access to that channel is very fast. Overall, random access to SEISAN waveform files is much faster with the binary read than the previous (version 5.0 and earlier) unformatted read. Only in the case where the whole file is read is the unformatted read faster.

<table>
<thead>
<tr>
<th>PC file structure</th>
<th>Sun and Linux file structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to and including version 6.0</td>
<td>PC structure from version 7.0</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>one byte: K indicates start of file</td>
<td>4 bytes: # of bytes following</td>
</tr>
<tr>
<td>one byte: # of bytes following</td>
<td>one block of data</td>
</tr>
<tr>
<td>128 bytes or less of data</td>
<td>4 bytes: # bytes in prev. write</td>
</tr>
<tr>
<td>one byte: # of bytes in previous record</td>
<td>4 bytes: # of bytes following</td>
</tr>
<tr>
<td>one byte: # of bytes in following record</td>
<td>one block of data</td>
</tr>
<tr>
<td>128 bytes or less of data</td>
<td>........</td>
</tr>
<tr>
<td>......</td>
<td>........</td>
</tr>
<tr>
<td>......</td>
<td>........</td>
</tr>
</tbody>
</table>

For 64 bit systems, the above 4 byte numbers are 8 byte numbers.
From version 7.0, the Linux and PC file structures are exactly the same. On Sun the structure is the same except that the bytes are swapped. This is used by SEISAN to find out where the file was written. Since there is always 80 characters in the first write, character one in the Linux and PC file will be the character P (which is represented by 80) while on Sun character 4 is P.
Appendix C

Response file formats used in SEISAN

SEISAN can use either SEISAN response format or GSE response format or SAC response format or RESP response format. The detailed SEISAN format is given here as well as the parts of the GSE format and SAC format used in SEISAN. Please see the SEED manual for details on the RESP response file format.

See also chapter 5.2.2

C.1  SEISAN response format

There are three SEISAN response formats. The instrument response can either be presented as (1) instruments constants, (2) pairs of frequency amplitude and phase or as (3) poles and zeros.

Line 1
  1    5: STATION CODE (A5)
  6   9: COMPONENT (A4), E.G. SZ L : SZ LOW GAIN, AN: ACCELERATION NS
 10   : CENTURY, 0: 1900, 1:2000
 11  12: YEAR
 14  16: DOY
 18  19: MONTH
 21  22: DAY
 24  25: HR
 27  28: MIN
 30  35: SECOND (F6.3)
 52  59: LATITUDE (F8.4), optional
 61  69: LONGITUDE (F9.4), optional
 71  75: ELEVATION (METERS), optional
 78   : P: Poles and zeros
      Blank: Seismometer period etc used for response info. See below for details.
      T: Use up to 30 tabulated values irrespective of what is given
APPENDIX C. RESPONSE FILE FORMATS USED IN SEISAN

below. If less than 30, blank characters must be given.

79: C: a combination of table, poles and zeros or instrument
    constants have been used, for information only. Value in 78
    must then be T.
    F: Force use of header response. Only
    gain at 1 hz is correct and 78 must be set to T.

Line 2
  1  80: COMMENT LINE DESCRIBING THE SYSTEM RESPONSE (A80)

OPTION 1

If character 78 is blank, option 1:

Line 3  
  1 - 80: (10G8.3) 1. SEISMOMETER PERIOD
           2. FRACTION OF CRITICAL DAMPING
           3. SEISMOMETER GENERATOR CONSTANT (V/m/s) or
              ACCELEROMETER SENSITIVITY (V/G)
           4. AMPLIFIER GAIN
           5. RECORDING MEDIA GAIN (I.E. 2048 COUNTS/VOLT)
           6. GAIN AT 1.0 HZ, UNITS: COUNTS/METER
           7. CUTOFF FREQUENCY FOR FILTER1 (HZ)
           8. # OF POLES FOR FILTER1 (NEGATIVE FOR HIGHPASS)
           9. CUTOFF FREQUENCY FOR FILTER2 (HZ)
          10. # OF POLES FOR FILTER2 (NEGATIVE FOR HIGHPASS)

Line 4  
  1 - 80: (10G8.3) FREQUENCIES AND #’S OF POLES FOR FIVE MORE FILTERS

Lines 5-13: RESPONSE CURVE in 3 blocks of 3 lines
  1 - 80: (10G8.3) FREQ.
  1 - 80: (10G8.3) AMPL. (REL. 1.0 HZ)
  1 - 80: (10G8.3) PHASE,

These are only for info in option 1

OPTION 2

If character 78 is T, use tabulated values, format as in option 1

OPTION 3

If character 78 is P, option 2:

Line 3  
  1 - 22: (1X,2I5,G11.4) 1. NUMBER OF POLES
C.2. GSE RESPONSE FORMAT

Below is an example of a GSE response file generated by RESP for the same parameters as used for the SEISAN file above. The GSE response format is rather complex, and can contain parts that will not be understood by SEISAN.

Below follows an example of the GSE response using FAP. The first line gives station and sensor type. The numbers following are gain in nm/c (0.15) at reference period (1. seconds), sample rate (20) and date. The following line (FAP2) gives a gain factor (1.) and the output units (V for Volts). Finally follows the frequency, gain and phase triplets.

Example of SEISAN FAP response file

KBS B Z100 1 1 1 0 0 0.000

Example of SEISAN PAZ response file using the same constants as above

KBS B Z100 1 1 1 0 0 0.000  P

C.2 GSE response format

Below is an example of a GSE response file generated by RESP for the same parameters as used for the SEISAN file above. The GSE response format is rather complex, and can contain parts that will not be understood by SEISAN.

Below follows an example of the GSE response using FAP. The first line gives station and sensor type. The numbers following are gain in nm/c (0.15) at reference period (1. seconds), sample rate (20) and date. The following line (FAP2) gives a gain factor (1.) and the output units (V for Volts). Finally follows the frequency, gain and phase triplets.
APPENDIX C. RESPONSE FILE FORMATS USED IN SEISAN

In the poles and zero representation, the file looks like:

```
CAL2 KBS B Z STS-1 0.15E+00 1. 20.00000 2003/ 1/ 1 0: 0
PAZ2 1 V 0.26000000E-05 2 3 Laplace transform
-0.12217305E-01 0.12464144E-01
-0.12217305E-01 -0.12464144E-01
0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00
0.00000000E+00 0.00000000E+00
DIG2 2 0.41900000E+06 20.00000 quanterra
```

The first line is the same as before. The PAZ2 lines has the following meaning: Normalization constant of seismometer and filter (0.26e-5), number of poles and zeroes (2 and 3), and the type of response (Laplace transform). The DIG2 line has the gain of amplifier and AD converter combined (0.419e6 c/V) and sample rate.

In the simplest case, the response is given by the PAZ and a scaling factor. It is common (like in SEED) to have two scaling constants, one that normalizes the PAZ to amplitude 1 at a calibration period, and another constant that gives the amplitudes in the physical units. This is NOT the case with the GSE2 format. The GSE2 response for PAZ normally contains at least two parts, the CAL2 line and a PAZ2 line. The scaling factor should scale the PAZ to output/input units, NOT normalize. In the CAL2 line, the system sensitivity at a calibration period is given in units input/output, but is generally not needed. The total response is given by the PAZ, multiplied with the PAZ2 scaling factor, or the product of several stages.

This is how SEISAN reads the response, however, if it finds that the PAZ2 gives normalized values at the calibration period, the response is multiplied with the sensitivity given in the CAL2 line (this is done because such GSE files have been seen).

FIR filters can be specified in GSE as an additional stage and can be written out by the RESP program. An example is given below. The FIR filter coefficients are required to completely describe the instrumentation. However, they are not used in SEISAN.

```
FIR2 3 0.10E+01 3 0.030 A 180
```
C.3 SAC format

Below is an example of the SAC response file format. Note that if zeros are zero, there is no need to write them. The response must be in displacement with unit meters.

```
ZEROS 3
POLES 4
-0.0123 0.0123
-0.0123 -0.0123
-39.1800 49.1200
-39.1800 -49.1200
CONSTANT 3.832338e+12
```

Note that the response file names must follow SEISAN response file names with SAC at the end like e.g. KEV_B_Z.2000-10-10-0000_SAC.

C.4 RESP format

The RESP files are obtained from a SEED file by the rdseed program when the R flag is set. See the chapter 41.3 for details.

Note: RESP files will not include all metadata stored in the SEED file, e.g. a rotation of the horizontal channels.
Appendix D

Some general subroutines in SEISAN in LIB

Subroutines in SEISAN are located in different files and it is not always easy to find what is available. Consequently there are some duplication. The following is a list of some of the common routines and their location. Very technical routines have not been listed. Some much used routines are not listed since they are found together in a file like routines for waveform handling in LIB/wave.for. There are also many routines in main programs which might be useful, they are not listed here. Most entries are subroutines but there are also a few functions. See also INF/seisan.all for a list of all subroutine files in LIB.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>r4sort</td>
<td>sorting</td>
<td>tau</td>
</tr>
<tr>
<td>r8sort</td>
<td>sorting double precision numbers</td>
<td>hypoloc1</td>
</tr>
<tr>
<td>hpsort</td>
<td>sorting</td>
<td>general</td>
</tr>
<tr>
<td>qsort</td>
<td>sorting</td>
<td>quarrycheck</td>
</tr>
<tr>
<td>mdian</td>
<td>calculate median</td>
<td>general</td>
</tr>
<tr>
<td>sei rand</td>
<td>random number generator</td>
<td>general</td>
</tr>
<tr>
<td>lin_interp</td>
<td>linear interpolation</td>
<td>general</td>
</tr>
<tr>
<td>lsqlin</td>
<td>linear regression</td>
<td>lsqlin</td>
</tr>
<tr>
<td>sdm</td>
<td>standard deviation</td>
<td>lsqlin</td>
</tr>
<tr>
<td>maxlik</td>
<td>maximum likelihood linear fitting</td>
<td>maxlik</td>
</tr>
</tbody>
</table>

Table D.1: Statistics
## Table D.2: Numerical

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>integrate</td>
<td>integration based on trapezoid rule</td>
<td>auto_proc</td>
</tr>
<tr>
<td>rtr</td>
<td>remove linear lsq trend</td>
<td>auto_proc</td>
</tr>
<tr>
<td>differentiate</td>
<td>differentiate</td>
<td>auto_proc</td>
</tr>
<tr>
<td>char_fu</td>
<td>calculate characteristic function</td>
<td>auto_proc</td>
</tr>
<tr>
<td>normalize_signal</td>
<td>normalize a signal</td>
<td>auto_proc</td>
</tr>
<tr>
<td>bcd</td>
<td>bcd byte to number</td>
<td>bcd</td>
</tr>
<tr>
<td>bndpas</td>
<td>butterworth filter, time domain</td>
<td>bndpas</td>
</tr>
<tr>
<td>recfil</td>
<td>many filter types, type domain</td>
<td>recfil</td>
</tr>
<tr>
<td>bworth</td>
<td>butterworth filter, freq. domain</td>
<td>sys_resp</td>
</tr>
<tr>
<td>pazresp</td>
<td>poles and zeros reponse</td>
<td>sys_resp</td>
</tr>
<tr>
<td>mpolys</td>
<td>multiply two polynomials</td>
<td>recfil</td>
</tr>
<tr>
<td>ctpoly</td>
<td>CONVERT ROOT DESCRIPTION TO POLYNOMIAL DESCRIPTION</td>
<td>recfil</td>
</tr>
<tr>
<td>bessel</td>
<td>generate bessel system function</td>
<td>recfil</td>
</tr>
<tr>
<td>delip</td>
<td>matrix normalization</td>
<td>delip</td>
</tr>
<tr>
<td>xy_ellipse</td>
<td>ellipsode error and the orientation from the north</td>
<td>err_ellipse</td>
</tr>
<tr>
<td>ellipse_x</td>
<td>major axis lengths and orientations from covariance matrix</td>
<td>err_ellipse</td>
</tr>
<tr>
<td>fft</td>
<td>fft</td>
<td>fft</td>
</tr>
<tr>
<td>reakft</td>
<td>fft real numbers ??</td>
<td>sig_spec</td>
</tr>
<tr>
<td>real2fft</td>
<td>two dim fft ??</td>
<td>sig_spec</td>
</tr>
<tr>
<td>four1</td>
<td>fft ??</td>
<td>sig_spec</td>
</tr>
<tr>
<td>dyadic</td>
<td>calculate dyadic of eigenvectors</td>
<td>fmap_subs</td>
</tr>
<tr>
<td>matsum</td>
<td>calculate matrix sum</td>
<td>fmap_subs</td>
</tr>
<tr>
<td>remove_dc</td>
<td>remove dc</td>
<td>remove_dc</td>
</tr>
<tr>
<td>rotate</td>
<td>rotate matrix</td>
<td>hyposub1</td>
</tr>
<tr>
<td>rotate</td>
<td>rotate a 3-vector represented in cartesian coordinates</td>
<td>hyposub1</td>
</tr>
<tr>
<td>scmuxmn</td>
<td>extremal values of the time series</td>
<td>sasubf</td>
</tr>
<tr>
<td>jacobi</td>
<td>eigenvalues and eigenvectors of a real symmetric matrix</td>
<td>sasubf</td>
</tr>
<tr>
<td>eigsr</td>
<td>sorts the eigenvalues</td>
<td>sasubf</td>
</tr>
<tr>
<td>eigen</td>
<td>eigenvalues and eigenvectors of a real symmetric matrix</td>
<td>hyposub2</td>
</tr>
<tr>
<td>GMPRD</td>
<td>matrix multiplication</td>
<td>focmec_exe_sub</td>
</tr>
<tr>
<td>intqf2</td>
<td>find eighe values and vectors</td>
<td>focmec_exe_sub</td>
</tr>
<tr>
<td>dec_to_bin</td>
<td>convert real number to binary</td>
<td>ga_lib</td>
</tr>
<tr>
<td>bin_to_dec</td>
<td>convert binary to decimal number</td>
<td>ga_lib</td>
</tr>
<tr>
<td>minv</td>
<td>svd of partial derivative matrix</td>
<td>hyposub2</td>
</tr>
<tr>
<td>polos</td>
<td>finds if point inside a polygon</td>
<td>polos</td>
</tr>
<tr>
<td>qdis</td>
<td>pythagorean distance calculation</td>
<td>quarrycheck</td>
</tr>
<tr>
<td>smooth</td>
<td>smooth data</td>
<td>sig_spec</td>
</tr>
<tr>
<td>prepare</td>
<td>pad and taper</td>
<td>sig_spec</td>
</tr>
<tr>
<td>taper</td>
<td>taper</td>
<td>sig_spec</td>
</tr>
<tr>
<td>precoh1</td>
<td>azimuth and vel. from 3 comp .</td>
<td>sig_spec</td>
</tr>
<tr>
<td>cross</td>
<td>cross correlation</td>
<td>sig_spec</td>
</tr>
<tr>
<td>crosscorr</td>
<td>——————, time domain</td>
<td>sig_spec</td>
</tr>
<tr>
<td>cor_time</td>
<td>——————, time domain</td>
<td>sig_spec</td>
</tr>
<tr>
<td>cor_freq</td>
<td>——————, freq ——</td>
<td>sig_spec</td>
</tr>
<tr>
<td>correl</td>
<td>correlation ???</td>
<td>sig_spec</td>
</tr>
<tr>
<td>rotate_comp</td>
<td>signal rotation</td>
<td>sig_spec</td>
</tr>
<tr>
<td>check_clipped</td>
<td>check if clipped signal</td>
<td>sig_spec</td>
</tr>
<tr>
<td>check4gap</td>
<td>find gaps in data</td>
<td>sig_spec</td>
</tr>
<tr>
<td>svbksbx</td>
<td>solves the equation A * X = B</td>
<td>ssvd</td>
</tr>
<tr>
<td>svdcmpx</td>
<td>———— complex ??</td>
<td>ssvd</td>
</tr>
<tr>
<td>svdvar</td>
<td>covariance of a matrix</td>
<td>ssvd</td>
</tr>
<tr>
<td>hubksbx</td>
<td>linear equation problem ??</td>
<td>ssvd</td>
</tr>
<tr>
<td>ludecmpx</td>
<td>———— ??</td>
<td>ssvd</td>
</tr>
<tr>
<td>pythag</td>
<td>pythagoras ??</td>
<td>ssvd</td>
</tr>
</tbody>
</table>
### Table D.3: Character handling

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEI UPC</td>
<td>convert any lower case to upper</td>
<td>libsei</td>
</tr>
<tr>
<td>upper</td>
<td>convert to upper case</td>
<td>quarrycheck</td>
</tr>
<tr>
<td>trimlen</td>
<td>characters in string, ignore blank</td>
<td>quarrycheck</td>
</tr>
<tr>
<td>trimbeg</td>
<td>first not blank char in string</td>
<td>quarrycheck</td>
</tr>
<tr>
<td>text_to_array</td>
<td>read numbers in text string into integer array</td>
<td>ga_lib</td>
</tr>
<tr>
<td>casefold</td>
<td>convert to capital</td>
<td>codecoutil</td>
</tr>
<tr>
<td>ucase</td>
<td>lower case to upper case</td>
<td>hyposub1</td>
</tr>
<tr>
<td>lcase</td>
<td>upper case to lower case</td>
<td>hyposub4</td>
</tr>
<tr>
<td>SEI LEFT</td>
<td>left justify a text string</td>
<td>libsei</td>
</tr>
<tr>
<td>SEI CLEN</td>
<td>length of text string</td>
<td>libsei</td>
</tr>
<tr>
<td>r_index</td>
<td>find character in text, starting search from last character</td>
<td>libsei</td>
</tr>
<tr>
<td>swap4</td>
<td>swap 4 byte data</td>
<td>swap</td>
</tr>
<tr>
<td>swap4_one</td>
<td>swap 4 byte data, one value</td>
<td>swap</td>
</tr>
<tr>
<td>swap2</td>
<td>swap 2 byte data</td>
<td>swap</td>
</tr>
<tr>
<td>swap2_one</td>
<td>swap 2 byte data, one value</td>
<td>swap</td>
</tr>
<tr>
<td>text_sort</td>
<td>sort text strings</td>
<td>text_sort</td>
</tr>
<tr>
<td>check whole</td>
<td>check string for spaces</td>
<td>isf_isc</td>
</tr>
</tbody>
</table>

### Table D.4: Input output see also rea and wave

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEI CMD PRO</td>
<td>get text from input</td>
<td>libsei</td>
</tr>
<tr>
<td>SEI CMD SET</td>
<td>get argument from command line, possibly execute it</td>
<td>libsei</td>
</tr>
<tr>
<td>SEI GET FILE</td>
<td>open file in current or alt. directory</td>
<td>libsei</td>
</tr>
<tr>
<td>SEI GET VALUES</td>
<td>get values from input string</td>
<td>libsei</td>
</tr>
<tr>
<td>SEI INTEGER</td>
<td>get integer from a text string</td>
<td>libsei</td>
</tr>
<tr>
<td>SEI OPEN</td>
<td>open file on next available unit</td>
<td>libsei</td>
</tr>
<tr>
<td>get_seisan_def</td>
<td>read SEISAN.DEF</td>
<td>libsei</td>
</tr>
<tr>
<td>file_ready</td>
<td>check if file is opened already and wait until it is closed</td>
<td>libsei</td>
</tr>
</tbody>
</table>
Table D.5: SEISAN specific

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>iasp91_filename</td>
<td>set name of iasp91 files</td>
<td>general</td>
</tr>
<tr>
<td>tomdir</td>
<td>get top dir name</td>
<td>general</td>
</tr>
<tr>
<td>get_env_op</td>
<td>get operator id</td>
<td>general</td>
</tr>
<tr>
<td>get_operator</td>
<td>———— interactively</td>
<td>eev_sub</td>
</tr>
<tr>
<td>get_env_event</td>
<td>get event name</td>
<td>general</td>
</tr>
<tr>
<td>put_env_event</td>
<td>put event name</td>
<td>general</td>
</tr>
<tr>
<td>put_env_base</td>
<td>put base name in memory</td>
<td>general</td>
</tr>
<tr>
<td>put_env_seistop</td>
<td>put seisan top</td>
<td>general</td>
</tr>
<tr>
<td>put_env_op</td>
<td>put operator id</td>
<td>general</td>
</tr>
<tr>
<td>get_env_base</td>
<td>get base name from ——</td>
<td>general</td>
</tr>
<tr>
<td>get_env_cal</td>
<td>get alt. CAL dir</td>
<td>general</td>
</tr>
<tr>
<td>get_def_base</td>
<td>get def. base</td>
<td>general</td>
</tr>
<tr>
<td>get_env_string</td>
<td>get any string</td>
<td>general</td>
</tr>
<tr>
<td>put_env_string</td>
<td>put any string</td>
<td>general</td>
</tr>
<tr>
<td>put_seisan_message</td>
<td>put a seisan message</td>
<td>seis_mes</td>
</tr>
<tr>
<td>get_seisan_message</td>
<td>get a seisan message</td>
<td>seis_mes</td>
</tr>
<tr>
<td>get_agency</td>
<td>get agency from ——</td>
<td>general</td>
</tr>
<tr>
<td>send_plot</td>
<td>send plot to laser</td>
<td>general</td>
</tr>
<tr>
<td>dir_char</td>
<td>directory separation char</td>
<td>general</td>
</tr>
<tr>
<td>uncompress</td>
<td>check if file is compressed and uncompress</td>
<td>general</td>
</tr>
<tr>
<td>sort_event</td>
<td>sort an s-file acor. to distance</td>
<td>general</td>
</tr>
<tr>
<td>get_agency_hyp</td>
<td>return agency in station file</td>
<td>general</td>
</tr>
<tr>
<td>get_baz</td>
<td>calculate station back azimuth for stations in waveform header</td>
<td>get_baz</td>
</tr>
<tr>
<td>component</td>
<td>convert from 2 to 4 char component</td>
<td>componen</td>
</tr>
<tr>
<td>findchan</td>
<td>find channel number corresponding to a station and component</td>
<td>findchan</td>
</tr>
<tr>
<td>filename</td>
<td>get filename seisan way</td>
<td>filename</td>
</tr>
<tr>
<td>init_networ</td>
<td>start UDP communication</td>
<td>network</td>
</tr>
<tr>
<td>send_udp_message</td>
<td>send UDP message</td>
<td>network</td>
</tr>
<tr>
<td>get_udp_message</td>
<td>receive UDP message</td>
<td>network</td>
</tr>
<tr>
<td>close_network</td>
<td>close UDP communication</td>
<td>network</td>
</tr>
<tr>
<td>ms_sleep</td>
<td>sleep function</td>
<td>network</td>
</tr>
<tr>
<td>number_of_events</td>
<td>count number of events in a CAT file</td>
<td>general</td>
</tr>
</tbody>
</table>
### Table D.6: S-files see rea for general routines

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>check_s</td>
<td>check s-file for errors</td>
<td>check_s</td>
</tr>
<tr>
<td>read_s_file_e</td>
<td>open and read s-file, close, for eev</td>
<td>check_s</td>
</tr>
<tr>
<td>write_s_file_e</td>
<td>write s-file, ————</td>
<td>check_s</td>
</tr>
<tr>
<td>auto_tr</td>
<td>get trace file names from s-file</td>
<td>auto_tr</td>
</tr>
<tr>
<td>auto_filename</td>
<td></td>
<td>autofil</td>
</tr>
<tr>
<td>findevin</td>
<td>finds an event in list of events</td>
<td>findevin</td>
</tr>
<tr>
<td>get_fstart</td>
<td>find start of s-file name and check name</td>
<td>findevin</td>
</tr>
<tr>
<td>merge_f</td>
<td>merge two s-files</td>
<td>merge_f</td>
</tr>
<tr>
<td>nortype</td>
<td>find if compact or normal d-file</td>
<td>nortype</td>
</tr>
<tr>
<td>rea_xxx</td>
<td>different routines</td>
<td>rea</td>
</tr>
<tr>
<td>sfilename</td>
<td>make s-file name</td>
<td>sfilname</td>
</tr>
<tr>
<td>statloc</td>
<td>find coordinates of a given station</td>
<td>statloc</td>
</tr>
<tr>
<td>statloc_many</td>
<td>———— of many stations</td>
<td>statloc</td>
</tr>
</tbody>
</table>

### Table D.7: Timing rotines

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstim</td>
<td>converts a data base timestring to abs time</td>
<td>abstim</td>
</tr>
<tr>
<td>etoh</td>
<td>epoch time to human time</td>
<td>sacsubj</td>
</tr>
<tr>
<td>htoe</td>
<td>calendar date to epoch time since January 1, 1970</td>
<td>sacsubj</td>
</tr>
<tr>
<td>dtoepo</td>
<td>convert julian date to epoch time</td>
<td>sacsubj</td>
</tr>
<tr>
<td>julian</td>
<td>calculates julian day</td>
<td>codecoutil</td>
</tr>
<tr>
<td>mnthdy</td>
<td>YEAR and DOY, return MONTH and DAY</td>
<td>sacsubj</td>
</tr>
<tr>
<td>systime</td>
<td>get system time</td>
<td>systime</td>
</tr>
<tr>
<td>datetimex</td>
<td>get date and time</td>
<td>systime</td>
</tr>
<tr>
<td>TIMSEC</td>
<td>time in year to double precision seconds after 1.1.0000</td>
<td>timerout</td>
</tr>
<tr>
<td>SECTIM</td>
<td>seconds after 1.1.0000 to date and time</td>
<td>timerout</td>
</tr>
<tr>
<td>DTE</td>
<td>day-of-year to date (month,day)</td>
<td>timerout</td>
</tr>
<tr>
<td>TIMDIF</td>
<td>time-difference in seconds of two two input dates and time</td>
<td>timerout</td>
</tr>
<tr>
<td>TIMADD</td>
<td>Adds seconds to date, and returns the new date</td>
<td>timerout</td>
</tr>
<tr>
<td>MONTH_DAY</td>
<td>number of days in a month from month and year</td>
<td>timerout</td>
</tr>
<tr>
<td>leap_year</td>
<td>find leap year</td>
<td>timerout</td>
</tr>
</tbody>
</table>
### Table D.8: General seismological

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>autoamp</td>
<td>peak to peak amplitude and period</td>
<td>autoamp</td>
</tr>
<tr>
<td>find_coda</td>
<td>find signal duration</td>
<td>autoProc</td>
</tr>
<tr>
<td>auto_coda</td>
<td></td>
<td>auto_coda</td>
</tr>
<tr>
<td>compute_brune</td>
<td>calculate brune spectrum</td>
<td></td>
</tr>
<tr>
<td>freq_event_type</td>
<td>determine if local or distance ev.</td>
<td>autoProc</td>
</tr>
<tr>
<td>m0def</td>
<td>decompose moment tensor into various representations</td>
<td>fmap-sub</td>
</tr>
<tr>
<td>pcnts</td>
<td>percentages of double-couple, CLVD &amp; iso in seismic source</td>
<td>fmap-sub</td>
</tr>
<tr>
<td>m0find</td>
<td>find scalar moment</td>
<td>fmap-sub</td>
</tr>
<tr>
<td>ptfind</td>
<td>find P &amp; T vectors</td>
<td>fmap-sub</td>
</tr>
<tr>
<td>mtrsce</td>
<td>set up moment tensor components from input moment tensor representation</td>
<td>fmap-sub</td>
</tr>
<tr>
<td>fpsol</td>
<td>calculate strike, dip &amp; slip of fault solutions from the P &amp; T vectors</td>
<td>fmap-sub</td>
</tr>
<tr>
<td>plaz</td>
<td>calculate plunge &amp; azimuth of eigenvectors in degrees</td>
<td>fmap-sub</td>
</tr>
<tr>
<td>mdcmp</td>
<td>calculate components of an arbitrary moment tensor</td>
<td>fmap-sub</td>
</tr>
<tr>
<td>AN2DSR</td>
<td>dip, strike and rake - A&amp;R convention from A and N</td>
<td>focmec_exesub</td>
</tr>
<tr>
<td>AN2MOM</td>
<td>moment tensor element from A and N</td>
<td></td>
</tr>
<tr>
<td>ANTPIN</td>
<td>other representations of fault planes</td>
<td></td>
</tr>
<tr>
<td>DSRIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PTTPIN</td>
<td>from T and P</td>
<td></td>
</tr>
<tr>
<td>GDMOT</td>
<td>ground displacement of incident P</td>
<td></td>
</tr>
<tr>
<td>grndmntn</td>
<td>vertical and radial ground disp. amp.</td>
<td></td>
</tr>
<tr>
<td>mb_att</td>
<td>mb attenuation curves veith et all</td>
<td>m_att</td>
</tr>
<tr>
<td>mb_att</td>
<td>-- gutenberg richter</td>
<td></td>
</tr>
<tr>
<td>plot_foc</td>
<td>plotting focal mechanism</td>
<td>plot_foc</td>
</tr>
<tr>
<td>fps_gap</td>
<td>find az gap in polarity data</td>
<td>plot_foc</td>
</tr>
<tr>
<td>fps_gap_all</td>
<td>find az gap in polarity and amplitude data</td>
<td>plot_foc</td>
</tr>
<tr>
<td>p_t_gap</td>
<td>calculate difference in P and T axis between two fault plane solutions</td>
<td>eeve_sub</td>
</tr>
</tbody>
</table>

### Table D.9: Coordinate handling

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>fold</td>
<td>geographic to geocentric coordinates</td>
<td>hyposub1</td>
</tr>
<tr>
<td>unfold</td>
<td>geocentric to geographic coordinates</td>
<td>hyposub1</td>
</tr>
<tr>
<td>cvrttop</td>
<td>convert from rectangular to polar coordinates</td>
<td>hyposub1</td>
</tr>
<tr>
<td>delaz</td>
<td>calculate the distance in km and azimuths</td>
<td>hyposub1</td>
</tr>
<tr>
<td>distaz</td>
<td>angular distance, azimuth and back-azimuth between two points</td>
<td>hyposub1</td>
</tr>
<tr>
<td>cart</td>
<td>geographical location to geocentric cartesian coordinates</td>
<td>hyposub1</td>
</tr>
<tr>
<td>geog</td>
<td>geocentric cartesian coordinates to a geographical location</td>
<td>hyposub1</td>
</tr>
<tr>
<td>latlon</td>
<td>Find a point on a sphere at a given distance and azimuth from an. point</td>
<td>hyposub1</td>
</tr>
<tr>
<td>azibazi</td>
<td>calculate distance, azimuth and back azimuth</td>
<td>azibazi</td>
</tr>
</tbody>
</table>
### Table D.10: Plotting, see seiplot for most general routines

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>fcirc</td>
<td>Plot a filled circle</td>
<td>plot_foc</td>
</tr>
<tr>
<td>draw_circle</td>
<td>draw a circle</td>
<td>plot_foc</td>
</tr>
<tr>
<td>draw_triangle</td>
<td>draw a triangle</td>
<td>plot_foc</td>
</tr>
<tr>
<td>xy_plot</td>
<td>general xy-plot</td>
<td>xy_plot</td>
</tr>
<tr>
<td>xy_plot_line</td>
<td>make a line in xy_plot</td>
<td></td>
</tr>
</tbody>
</table>

### Table D.11: Waveform handling, see wave for most general routines

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>seisinc</td>
<td>read a seisan format file</td>
<td>seisinc</td>
</tr>
<tr>
<td>shead</td>
<td>make seisan waveform channel header</td>
<td>shead</td>
</tr>
<tr>
<td>sheads</td>
<td>——main and channel headers</td>
<td>sheads</td>
</tr>
</tbody>
</table>

### Table D.12: Routines for logging

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>get_log_file_name</td>
<td>This subroutine creates the name of the log file from the name of the sfile</td>
<td>seisan_loggi</td>
</tr>
<tr>
<td>create_log_file</td>
<td>This subroutine creates a log file (is it does not exist), name=sfilename.LOG</td>
<td>seisan_loggi</td>
</tr>
<tr>
<td>show_log_file</td>
<td>This subroutine sends the content of the logfile to the screen</td>
<td>seisan_loggi</td>
</tr>
<tr>
<td>sfile_to_log_file</td>
<td>This subroutine adds the sfile to the end of the log file</td>
<td>seisan_loggi</td>
</tr>
<tr>
<td>move_log_file</td>
<td>This subroutine add actions to the log file</td>
<td>seisan_loggi</td>
</tr>
<tr>
<td>get_log_time</td>
<td>This subroutine returns a string used to timestamp the log file</td>
<td>seisan_loggi</td>
</tr>
<tr>
<td>message_to_log_file</td>
<td>This subroutine added a message in the logfile</td>
<td>seisan_loggi</td>
</tr>
<tr>
<td>pick_to_log_file</td>
<td>This subroutine adds the character used to pick phases etc.</td>
<td>seisan_loggi</td>
</tr>
<tr>
<td>string_to_log_file</td>
<td>This subroutine add a string to the logfile</td>
<td>seisan_loggi</td>
</tr>
</tbody>
</table>
APPENDIX D. SOME GENERAL SUBROUTINES IN SEISAN IN LIB
Appendix E

Changelog

This appendix list the changes that have been added to SEISAN since version 8.2. Please note that most changes are also notes in the header of the source code.

E.1 SEISAN version 10.3 (2015-01-17)

Changes since version 10.1:

- Latest version of HYPOINVERSE now works
- New version of CODAQ which can work with several data sets

NOTE: Version 10.2 was only distributed under Windows so most changes to version 10.2 are also changes to 10.3

E.2 SEISAN version 10.2 (2014-09-20)

Changes since version 10.2:

- Changes to program AUTOMAG to refine grid search
- New program GETSTRESSDROP to extract and analyse stress drops for S-files
- Seisbrain and Seisan tutorial now include SeisanExplorer
- SeisanExplorer can make a simple waveform plot
- SeisanExplorer plots time of day distributions, see page 58
- Program LSQ can now also plot histograms
- Program P_ALIGN can now time shift with respect to both P or S phases, see page 270
- A few minor bugfixes
• A major problem with SE and spectral values fixed
• Latest version of Hypoinverse now works

Note: Version 10.2 was only released for Windows.

E.3 SEISAN version 10.1 (2014-06-19)

Changes since version 10.1:

• Plot picture files from EEV with command pp. Try with earthquake on June 3rd 1996 in TEST base. Files are in PIC or local directory. A new linetype P is used in S-file for the file name. In SEISAN.DEF PLOT_PICTURE,COMMAND must be set.

• Plot pdf files from EEV with command pp. Files are in PIC or local directory. A new linetype P is used in S-file for the file name. In SEISAN.DEF PLOT_PDF,COMMAND must be set.

• A new command em in EEV will edit an ISO file listed in the S-file.

• A new program plotspec, called from EEV with command ps, will plot all spectra made with AUTOMAG in EEV. Bad fits can be deleted interactively.

• MULPLT will select possible arc channels before reading them.

• Q correction in most programs have been changed to optionally be frequency independent set q0 for f below a given frequency, usually 1 Hz.

• New Perl script (textttmerge_seisan.pl) to associate CAT files, many new options compared to ASSOCIATE.

• Program LSQ can now also plot curves from xy-data.

• A new program ISCCSVC2NOR to convert ISC CSV catalog data to Nordic format.

• WAVETOOL: new option to convert station and channel names of miniseed files

• Data in SeisComP archives with quality make M, R or E, is now read, if D is not present

E.4 SEISAN version 10.0 (2013-06-11)

Changes since version 10.0:
The main change is the SeisanExplorer has been more integrated with SEISAN and can be used with many new functions. Other changes are:

• New program to make automatic spectra and amplitudes, AUTOMAG, also used in EEV, based on AUTOSIG.

• New program: RDSEED2SEISAN - use it to get SEED metadata into SEISAN

• Program DELS to delete part of S-file, like all S-readings. Also option form EEV
• MULPLT can use arrows to scale amplitudes

• Options in EEV to plot Ml as a functions of distance and seismic moment as a function of arrival time

• New and improved CLUSTER program

• New program ASSO for events association, also use magnitude as a criteria, mainly for seismic hazard

• R command in EEV can now be used to rename event type (L, R or D) and to change event ID e.g. LE or LP (use LB for blank)

• All programs using the REA data base can now work back to year 0000

• MULPLT changes: new parameter in MULPLT.DEF to fix the filter, set a center station for muplt_area option, set output format for Out option

• New options in EEV-MULPLT to deal with archives, particularly the use of wild cards. In S-file there is no longer a differentiation between, SCP and BUD, both are now called ARC and the archive type is given in SEISAN.DEF

• filenr.lis file now up to 99999 files

• New sample program sample_resp_correction

E.5 SEISAN version 9.1 (2013)

Changes since version 9.1:

• Chad Tabant’s seed read and write routines are now used in all programs.

• Dregers moment tensor inversion integrated with SEISAN.

• Compilation has been simplified with the integration of DISLIN and Libmsed into the seisan distribution.

• New option in SPEC(no plot, used for batch).

• New conversion program HINOR, Hypoinverse archive format to NORDIC.

• Several programs in SEISAN reads Guralp gcf format.

• Several programs in SEISAN read Helberger format (used for MT inversion).

• MULPLT can now, for the first time, write out any processed signal seen on the screen (option OutW).

• Filter routines gave been reorganized giving more options (see MULPLT section on filters).

• MULPLT can overlay channels.

• MULPLT can select many channels in multitrace mode by left and right button mouse click.

• MULPLT can plot files in a file of files via EEV, LIST option.
• MULPLT can select only to plot stations for plotting in a given distance from a point, usually the epicenter.
• SELECT has a new option for selecting data for CODAQ.
• GMAP new option: automatic plotting of epicenters in Google Earth, see page 215.
• GMAP: Error ellipses are implemented.
• CODAQ output results in at station-event midpoints
• CODAQ_area: new program to sort CODAQ output in areal bins.
• JSEISAN has been removed.
• SeisanExplorer, a new graphical interface to replace EEV and JSEISAN, distributed separately.

E.6 SEISAN version 9.0.1 (2011)

Changes since version 9.0.1:
• Fixed some minor bugs in version 9.0.
• New options and changes in EEV: IFP: Input manually fault plane solution, INPUTFPS: input complete fault plane solution line, COML: Input geographical location line, COMF: Input felt event comments, U: Update event (was UPDATE, UP: Update event list (was U), M: Input model indicator on header line.
• program FOC now also makes rose diagrams.

E.7 SEISAN version 9.0 (2011)

Changes since version 9.0:
• New compiler and graphics system for Windows: This is the largest change. Windows now uses Gfortran and gcc and the graphics library is DISLIN. This has stabilized the graphics on Windows end enabled to use the same compiler on most platforms. However, the gfortran on Windows has created new problem in some programs most of which probably has been solved.
• SEISAN can now extract and plot data from a BUD and a SEISCOMP archive, read more at page 14
• EEV has several new options: CM: Copy many files, DD: Duplicate header, FH: HASH fault plane solution, FP: FPFIT fault plane solution, FO: Plot all fault plane solutions, IFP: Input manually fault plane solution, INPUTFPS: input complete fault plane solution line, COML: Input geographical location line, COMF: Input felt event comments.
• FOC: A program to plot and analyze fault plane solutions.
• FPFIT: FPFIT fault plane solution program with SEISAN driver program FPFIT_SEISAN.
• HASH_SEISAN: HASH fault plane solution program.
• THIS PROGRAM IS NO LONGER PART OF SEISAN (2018): GETPDE - A new program that grap the PDE from the USGS web page and add the events in a database.
• SEIS2VIEWER - A new program for plotting earthquakes on a map, command smap, see page 224.
• EPIMAP: Can now plot fault plane solutions.
• MULPLT: Has a three component option facilitating working with three component data.
• AVQ: new program to average Q-relations.
• MAG2: new program to invert amplitudes for ML scale.
• Individual size of graphics windows for different programs can now be set in COLOR.DEF.
• The following programs do not work with Gfortran: HYPINV, HYPJSC, NORHYP, ARCSEI

E.8 SEISAN version 8.3 (2010-06-05)

Changes since version 8.3:
• Manual now written in Latex, html version is available
• Unix setup files were remamed to SEISAN.csh and SEISAN.bash to make them visible
• MULPLT: Plot hour and minute on time axis, show shortcut keys on menu, select picked traces from trace selection, new keyboard shortcuts when reading amplitudes, also shortcut keys for the first 10 traces in trace selection (these changes were put in by Wayne Crawford)
• Two events are now included in the SEISAN software, so that one can plot data after unpacking SEISAN, both events are found by typing ee 199606 TEST (Note : in unix one must first source the COM/.SEISAN file)
• PINV, new program for estimation of fault plane solution using polarities, see page 347
• 64 bit, have only been tested on Linux
• SEISAN can now use SAC PAZ response as created with rdseed
• code now compiles with gfortran
• AUTOREG has new option for moving waveform file to WAV, see page 256
• EEV accept SEED orientation codes : A,B,C,1,2,3,U,V,W,S and Z,N,E
• MULPLT orientation code 1 and 2 are read as N and E, respectively. 1 and 2 are used if orientation is different from N and E. (This still requires a more general solution)
• ASCSEI, A bug in in the reading of the input file was found and fixed, the first sample was lost if data was not PSN data.
• ISCNOR : A bug in the reading of surface wave amplitudes was fouled and fixed.
• The problem with extracting time windows on the last page in MULPLT continous mode has been fixed, page 149
• EEV copies the name and path of the current s-file to a file named `eev.cur.sfile', when the system command 'o' is used.
• Plot STATION?.HYP and SEISAN polygon files with Google Earth using GMAP, see page 215.
• The Herrmann modeling programs finally work under Windows.
• FOCMEC: Use of amplitudes has been improved and bugs fixed in amplitude section.
• New broadband body wave and surface wave magnitudes have been implemented.
• New conversions programs: AHSEI, DIMASSEI, DATABASE2MSEED
• Noise spectra from continuous data, CONNOI and EVANOI.
• Magnitude implementation has been adjusted to the new IASPEI standard, see table below.

<table>
<thead>
<tr>
<th>Magnitude</th>
<th>Old SEISAN</th>
<th>ISC</th>
<th>Old IASPEI</th>
<th>NEW IASPEI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td>AMP AMPL</td>
<td>AML</td>
<td>IAML</td>
<td>IAML</td>
</tr>
<tr>
<td>mb</td>
<td>AMB</td>
<td>IAmB</td>
<td>IAML</td>
<td></td>
</tr>
<tr>
<td>mb</td>
<td>AMP AMB AMpb AMb AMPB</td>
<td>IAmB</td>
<td>IVmB_BB</td>
<td></td>
</tr>
<tr>
<td>Ms</td>
<td>AMP</td>
<td>AMS</td>
<td>IAMS20</td>
<td>IAMS20</td>
</tr>
<tr>
<td>MS</td>
<td></td>
<td>IAMSBB</td>
<td>IVMs_BB</td>
<td></td>
</tr>
</tbody>
</table>

E.9 SEISAN version 8.2.1 (2008-10-10)

Changes since version 8.2.1:

• Improved handling of continuous data bases, particularly with SEED data
• New and expanded training document
• Several bug fixes reported after version 8.2
• Mistakes in manual reported after releases of version 8.2
• More examples of making response files
• Improved filtering in MULPLT by introducing tapering

E.10 SEISAN version 8.2 (2008)

Changes since version 8.2:

• Improved SEED reading and writeing (still not perfect, sometimes problem with Steim2)
• SEED channel naming convention now used
• SAC reading and writing also under windows
• New conversion programs
- Improved WAVETOOL
- SEISEI use all formats for input and SEISAN and MiniSEED for output
- Store waveform data in memory for faster plotting (see MULPLT section, page 143.
- Array processing of teleseismic P- arrival on regional network using plane wave approach, PFIT
- Plot of arrival times using EEV
- Epicenter plotting using GoogleMap or GoogleEarth (program GMAP)
- Spectral analysis also of teleseismic events
- Particle motion plot in MULPLT
- In MULPLT from continuous plot with one channel, it is possible to extract out time windows in a data file, page 149.
- SEISAN has been tested on Vista and there are problems with some graphic programs like LSQ.
- EDRNOR, conversion program for USGS parametric data
- GSERESP2SEED, program to create dataless SEED volumes from GSE response files, using GSE2SEED
- QLg program has new features to test effect of noise and source perturbation
- Instructions on how to use SEISAN under Cygwin have been added
Index

Q_LG, 437
2 column ASCII WAV file, 298
3-component stations, 111, 305
64 bit computer, 20, 22, 434
64 bit computer, 556
523, 293, 528
Add waveform format, 12
AFAD, 287
AFADSEI, 287
Aftershock processing, 328
AGA, 11
Agency, 21, 126, 274, 462
Agency, select for, 231
AHSEI, 287, 292
Align traces, 269
All channel mode, 161
Amplitude, 165
Amplitude attenuation, 160
Amplitude phases, 164
Amplitude, automatic, 165
Andalucia, 274, 286, 295
Angle of incidence, 338
Apparent velocity, 97, 129, 165, 169, 339
APPEND, 256
Append another event to current event, 92
ARC_ARCHIVE, 15
ARC_CHAN, 15
ARCFETCH, 287
Archive, channel specification, 528
Archive, define channels, 15
Archive, extract, 263
ARCSEI, 287
Arceisc def, 287
Array processing, 493
Array stations, 111
array.out, 98
ARX, 265
ASCII files, 267, 290
ASCII waveform file, 2 columns, 298
ASCII waveform format, 284
ASCII, convert to, 290
ASCII, convert to SEISAN, 290
ASCII, SEISAN to, 267
ASCSEI, 290
ASSO, 411
ASSOCI, 409
Associate events in time, 100
AUTO, 308
auto.log, 308
auto.out, 308
AUTO_LOCATE, 194
AUTO_PROCESS, 194
Automag, 319
AUTOMAG.list, 325
Automate EPIMAP, 206
Automatic, 305
Automatic amplitude, 165
Automatic amplitude determination, 319
Automatic coda length, 166, 194
Automatic phase picking, 101, 305, 310
Automatic polarity reading, 338
Automatic processing, 308
Automatic spectral level determination, 319
Automatic spectrum, MULPLT, 177
Automatic amplitude reading, 337
AUTOPHASE, 310
AUTOPIC, 26
AUTOPIC.INP, 305
Autopic.out, 305
AUTORATIO, 337, 365
AUTOREG, 256
Autoscale depth profile, 204
Autoscaling, 171
Autosig, 315
Autosig.par, 317
AUTUPHASE, 338
Average Q-relations, 407
Average station residual, 128
AVQ, 322, 407
Azimuth, 165, 305
Azimuth error, 128
B-value, 331
Back azimuth, 97
Background noise, 181, 446
Band reject filter, 196
Bandpass filter, default in mulplt, 194
BASE, 85
Bash shell, 20
BGISEI, 291
Binary SEISAN waveform file structure, 559
Bisquare weighting, 128
Body wave magnitude, 115
BOUCH, 123
Bouch.out, 129
BOUSEI, 123
Brune spectrum, 529
BUD archive data, 14
BUL, 247
BUL.INP, 247
Bul.ps, 248
Bullen table, 140
Bulletin, 247
Bulletin program, 248
BVALUE, 331
Bvalue.eps, 333
Bvalue.out, 333
c++, 528
CAL directory, 294, 499
Calibration file, 505
CAT directory, 12
CAT-file, 11, 17, 67, 131
CAT_AGAA, 274
Cat_aga.par, 274
Catalog work, 12, 25, 103, 104, 242, 252, 409
Catday.out, 252
Cathour.out, 252
Catmonth.out, 252
CATSTAT, 252
Catyear.out, 252
Change event id, 100
Change of day, 132
CHANGE_RESP_NAME, 514
Channel order, 161
Channel selection, 145, 155
Check, 531
Check data base, 85
Check polarity, 338
CHECK_BASE, 88
China, 291
Chronological order, 243
CITSEI, 291
CityShark, 291
CLUSTER, 456
CNVSSR and CNVSSA, 293
Coda length, 128, 164, 166, 194, 459
Coda magnitude, 114, 128, 230
CODAQ, 383
codaq summary, 387
CODAQ select events, 231
Codaq.area.nor, 392
Codaq.imp, 383
Codaq.par, 383
CODAQ_AREA, 393
CODAQ_POLYGON, 401
COLLECT, 239
COLOR.DEF, 26
Command to the operating system, 97
Commands in EEV, 87
Comments in S-file, 95
Common tasks in SEISAN, 68
Compact file, 17, 242
Compare hypocenters of two files, 418
Compare magnitudes of two files, 418
COMPARE_HYP, 418
Compile programs, 28
Compiling SEISAN, 28
Component for, 165
Component codes, 13
Component in MULPLT, 165
Component names in S-file, 165
Components in MULPLT and EEV, 147
Composite fault plane solution, 349
Compression, 13
CONDDET, 328
CONGAP, 257
connoi, 257
Conrad interface, 127, 128
Consistency check, 129
Cont, 162
Continuous data, 16, 149, 162, 267
Continuous data base, default selection, 36
Continuous plotting, 149
Contour file, 205
Conversion of component codes, 284
Conversion of station codes, 284
Conversion programs, 273
Convert angles, 338
Convert to ASCII, 296
Converted magnitude, 462
Copy events, 93
COPY_WAV_DIR, 256
Corner frequency, 180
CORR, 421
Correct polarity, 269
Correlation, 421
cpp, 528
Crash, check base, 85
CRISIS2012, 154
Crustal model, 26
Crustal parameters, 439
Crustal structure determination, 122
CSS format, 291
CSV format, 274
CSVNOR, 274
Curve, plot, 253
Cut MiniSEED files, 265
Daily number of events, 252
Damped signal, 339
DAT directory, 26
Data base content, 85
Data base error, 94
Data base check, 85
Database generation, 26
Database name, 26
Database security, 84
Database structure, 103, 104
DATABASE2MSEED, 260
Dataseis, 293
DATASELECT, 77
DC level, 156
Default channel, 192
Default continuous data base, 36
Default filter, 194
Defaults, 192
DEL directory, 0
DELET database, 93, 103
Delete amplitudes, 292
Delete automatic picks, 163
Delete event, 85, 93
Delete many S-files, 246
Delete P-phases, 292
Delete part of S-file, 292
Delete S-file, 163
Delete spectra, 326
Delete WA amplitudes, 326
Delete waveform file, 163, 170
DELF, 245
DELS, 292
Depth as function of rms, 100
Depth profile, 204
Depth, fixing, 120
Depth, select for, 231
Depth, start value, 120
Different databases, 103
Digital data, 76
Dimension, 22
Diminution function, 175
Dip, 341
DIRF, 245
Dislin, 31
Distance id, 230
Distance indicator, 162
Distance order, 162
Distance range, plotting, 161
Distance range, select for, 231
Distance weighting, 118
Double difference location, 477
DR3024 and DR3016, 292
Dreger, 369
DRMS, 121
DRSEI, 292
Drum recording, 149
Duplicate event, 93
Duplicate ID, 54, 242
Duration magnitude, 128
Earthquake catalogue, 451
Earthquake statistics, 251, 252
EDDBPICK picking program, 278
Editor, 21
EDRNOR, 274
EEV, 67, 86
EEV over several months, 94
EEV print command, 99
EEV, jump to another month and database, 96
EEV.OUT, 93
Eliminate outlier, 119
email, 93
End time, 26
Environmental variable, 21
Epicenter area selection, 206
EPIMAP, 26, 201, 250
Epimap.cor, 206
Epimap.inp, 206
INDEX

Epimap.num, 206
EQRISK, 152
Error estimate, HYP, 129
Error in S-file, 94
Errors in hypocenter, 231
EUROPE.MAP, 26
evanoi, 257
Event detection, 328
Event felt, select for, 230
Event ID, 162
Event, duplicate ID, 242
EXFILTER, 469
Explosion, 94, 104, 163, 230
Explosion filtering, 469
Extract data in MULPLT, 171
Extract from archive, 263
Extract from waveform file, 171
Extract waveform files, 260, 265
Extracting events from the database, 239
Fault plane solution, 94, 351
Fault plane solution, plot, 209
Fault plane solution, plotting several, 340
Fault plane solution, plot, 211
FDSN, 296
Federation of Digital Seismographic Networks, 296
Felt information, 517
Felt information in S-file, 547
File conversion and modification programs, 273
File number, 144, 145
File types, 17
FILENR.LIS, 18, 245, 501
Filter and spectral analysis, 156
Filter, number of poles, 156
Filters in MULPLT, 155
Fix filter in mulplt, 194
Fixed scaling, 171
Fixing depth, 120, 128
Fixing location, 120, 128
Fixing origin time, 120, 128
FK analysis, 169, 193
FOC, 354
FOCMEC, making a synthetic input for testing, 341
Focmec.eps, 341
Focmec.inp, 338
focmec.lst, 341
Focmec.out, 341
Format, mulplt, 146
Format, Nordic, 557
Format, volcano information, 489
Format, waveform, 551
FPFIT, 342
Güralp, 292
Gain factor, 553
Gap range, select for, 232
Gaps in Seed, 143
GBV recorder, 293
GCF format, 292
gcf format, 12
Generic Mapping Tools, 283
Geometrical spreading, 175, 161
GeoSig, 293
GeoSys, 293
Geotech Instruments, 294
GET_ARC, 263
get.arc.out, 263
GET_STAT, 127
Get_wav.out, 265
GET_WEB_LOCATIONS, 81
GhostView, 24
GIINOR, 275
GIISEI, 292
Global event location, 117
Global station coordinates, 283
GMAP, 94
Gmap, 219
GMAP advanced, 216
GMAP automatic, 219
GMAP SEISAN.DEF parameters, 38
GMAP simple, 215
GMT, 105, 211, 283, 529
GMT, export fault plane solution, 351
GMTXY, 105, 253, 529
Go to event by number or date, 92
Google Earth, 26, 211, 215
Google map, 215
Graphics, 105, 229
Graphics problem on PC, 25
Graphics window, size, 27
Gridsearch.out, 205
Ground displacement, 161
GSE, 12, 501
GSE format, 282
GSE response file format, 563
GSE to Nordic, 279
GSE2.0, 12
GSERESP, 203
GSN, 293
GSRSEI, 293
GURALP, 12
GURSEI, 292

H_model.par, 122
H_model.out, 122
Hard copy files, 21
Hardcopy in single trace mode, 161
Hardwired constants, 189
HARNOR, 275
Harvard moment tensor solution, 275
HASH, 344
Header times, 269
Helmberger format, 12
Herkij_distance, 37, 193
Herrmann, 71, 423
HERSEI, 423
High accuracy, 37, 55
High pass filter, 156
HINNOR, 275
HINOR, 275
Hints on modelling, 428
Histogram, plot, 255
Homepage, 7
Horizontal component, 159
HYP output in S-file, 124
HYP problem, azimuth, 111
HYP, do not locate, 117
HYP, multiple model mode, 122
Hyp.cat, 243
HYP_ISC, 140
HYPINV, 138
HYPNOR, 275
HYPO style weight, 105
HYPO71, 95, 125, 133
HYPO71 files to Nordic files, 275
Hypocenter program, 111
Hypocenter, add to S-file, 96
Hypocenters, more than one, 103
HYPO2DD, 522
hypoDD, 477
Hypo_inverse, 95, 137
Hypo_inverse format, 282
Hypo_inverse to Nordic, 275
Hypsum.out, 131

IASP91.HED, 13, 434
IASPEI phases, 111, 129
IASPEI Seismic Format, 278
IASPEI software library, 294
IASPEI91, 117, 129
Iceland, 298
ID-line, 111, 103
IL, 96
iLoc, 141
IMS to Nordic, 279
IMSI.0, 278
IMSNOR, 129
Index file, 18, 87, 93, 103, 104, 123
index.codaq, 231, 386
index.out, 233
Index.codaq, 93, 385
INPUTTEPI, 96
INPUTTEPI, 96
Inserting events in the database, 211
Instrument correction, 156
Instrument response, 177, 501
Interactive work, 85
International Monitoring System, 278
Inversion, 473
Inversion, 476
INVRAD, 96, 130
IRIS, 293, 296, 298
IRISEI, 293
ISC, 79, 118, 275, 277–279
ISC data to Nordic, 279
ISC location, 96, 140
ISC to Nordic, 275, 279
Isloc.def, 140
ISCI.0, 275
ISCI.0, 278
ISFNOR, 278
ISFNOR, 278
ISMSEI, 293
Israel, 274, 279
Java, 528, 530
JHD, 473, 476
Joint hypocenter determination, 473, 476
K2SEI, 293
KACSEI, 293
Kappa, determine, 443
Key, weight, 153
Kinematics, 278, 293
INDEX

KINSEI, 293
KW2ASC, 293

Least squares, 253
Lennartz, 294, 296
LENPCQ, 294
LENSEI, 294
Lg phase, 111
Lg phase velocity, 128
Lg waves, 437
Linux file structure, 557
LIST file in MULPLT, 147
Local database, update, 243
Local index file, 103, 104
Local magnitude, 112, 230, 461
Local magnitude parameters, 129
Local minimum, 121
LOCAL_CAL, 211, 499
LOCALITY in S-file, 547
Locate event, 96
Locate two events together, 96
Locate with Bullen table, 140
Locate with ISC program, 96, 140
Location code, 111, 127, 284
Location code in HYP, 132
Location code in MULPLT, 165
Location programs, 111
Location with one station, 111
Location, max distance, 128
Location, min # of stations, 129
LOG directory, 11
Log file, 243, 244
Logging, 105, 150
Lomax, 225
Long phase names, 117, 542
Low pass filter, 156
Lower bound magnitude, 158
LSQ, 253, 529
Lsq,eps, 228
Lsq_gmt.out, 253

M88SEI, 294
MacOSX, 23, 30
MACROMAP, 517
Macromap.ps, 518
Macroseismic information, 96, 517
MAG, 180, 250, 459
MAG2, 463
Mag_amp.out, 462
Mag_coda.out, 462
Mag_mag.out, 462
Mag_new.out, 462
Mag_newa.out, 462
Mag_spec.out, 462
Magnitude, 112, 128, 129, 459
Magnitude conversion, 140
Magnitude correction, 131
Magnitude in epimap, 202
Magnitude order, 250
Magnitude parameters, 129
Magnitude relation, 159
magnitude relations, 73
Magnitude residuals, 126
Magnitude storage, 72
Magnitude weight, 127
Magnitude, fixed, 117
Magnitude, move from 2 header to first, 250
Magnitude, plotting, 73
Magnitude, select for, 231
Magnitudes without type, 230
Magnitudes, more than 3, 103, 117
MAGSTAT, 467
Mailing lists, 7
Makefile, 28
MAKEREA, 26, 256
Manipulating magnitudes, 72
Map files, 26, 283
MAPG, 411
MARS88, 294
Maximum expected magnitude, 453
Maximum likelihood, 253
Maximum likelihood b-value, 333
Maximum no of iterations, 128
Maximum number of traces, 22
Maximum observed magnitude, 453
mB, 230
mb, 116, 230
Mc, 114, 459
Memory for waveform data, 143
Merge catalogs, 103
Merge events near in time, 109
Merge events near in time and distance, 411
Merge waveform files, 170, 268
MERGE_S, 410
MERGE_WAVEFORM, 171
Merging files from different networks, 268
MiniSeed reading, 149
ML, 112, 129, 459, 460
Model indicator, 163
INDEX

MODEL.DEF, 121
Modelling parameters, Q, 127
Modes of operation, 67
Moho, 127, 128
Moment, 180, 443
Moment magnitude, 116
Moment tensor inversion, 351, 369
Moment tensor, plotting, 351
Monthly epicenter files, 243
Monthly number of events, 252
MOPAD, 351, 357
Mouse, test, 529
Move magnitudes in S-file, 73
Moving files between Sun and PC, 18
MS, 230
Ms, 115, 230
MSCUT, 265
MULPLT, 26, 76, 99, 143
MULPLT WAV OUT FORMAT, 194
MULPLT, automatic spectrum, 177
Mulpkt, format, 146
MULPLT, LIST file, 147
Mulpkt, option cont, 192
MULPLT, output waveform format, 171
MULPLT, plot distance, 156
MULPLT, plotting many files from EEV, 147
MULPLT, plotting station in a max distance, 161
MULPLT.DEF, 26, 102, 189
MULPLT.DEF in SE, 42
Mulpkt. ext, 149
MULPLT.OUT, 182
mulplt.wav, 171
MULPLT: Orientation code, 145
Multiple event S-file, 241
Multiple users on Linux/Unix, 22
Multiplexed files, 76
Multitrace mode, 146
Mw, 116, 180

Nakamura, 143
Name of the local Index file, 105
Nanometer, 112
Nanometrics, 294
NANSEL, 294
NDC, 278
Nearest station, 120
NEIC, 294
NEISEI, 294
Network code, 14, 127, 284
Network code in HYP, 132
network code in MULPLT, 165
Network code, put in waveform file, 284
New events into SEISAN, 162
NEWAVE, 78
Noise spectrum, 178, 181, 143, 145
Noise study, 181, 446
NOR2DD, 177
NOR2QML, 282
nor2qml, 290
NORCSY, 281
Nordic format, 11, 17, 537
Nordic to HYPO71, 282
NORGSE, 282
NORHEAD, 73, 250
NORHIN, 282
Northerz-Westphalia, 286
NORHYP, 282
NORMS, 17, 278
Normalization constant, 505
NRWSEI, 286
Number of poles, filter, 156

ObsPy, 25
Operator ID, 243
Order of magnitudes, 250
ORFEUS, 287
Orientation code in MULPLT, 145
OSOSEI, 294
Other databases, 11
Outlier in HYP, 119
Output, corrected trace, 171
Overlay channels in MULPLT, 171

P phase, 111
P-phase picking, 315
P ALIGN, 269
Parameter files, 26
PCEQ format, 294
PCK files, 278
PCQSEI, 294
PCSUDS, 298
PDAS files to SEISAN, 294
PDASEI, 294
PDE, 282
PDE bulletin file, 274
PDE e-mail, 282
PDE2DOR, 282
PDF files, 99
Peak ground acceleration, 452
Peak ground velocity, 452
INDEX

Period, 165
Period for amplitude, 72
Perl, 529
Peterson noise model, 131, 443
PFIT, 97
Pg phase, 111
PGA, 452
Phase mouse key, 192
Phase name, 542
Phase name key, 192
Phase picking, 69, 143, 164
Phase picking mode, 161
Phase reading indicator, 145
Phase weight key, 192
Phase with amplitude, 164
Phase, compare theoretical, 95
Phase, select in data base, 231
PIC directory, 105
Picking phases, use of filters, 155
Pictures, 99
Pictures, plotting, 105
PINV, 347
PITSA, 296
Place names, 204
Plot a curve, 253
Plot a histogram, 253
Plot contours, 201
Plot distance in MULPLT, 156
Plot fault plane solutions, 211
Plot PDF file, 38
Plot PDF files, 99
Plot picture, 38
Plot picture files, 99
PLOTFOC, 357
PLOTPOLARITY, 338, 361
PLOTRATIO, 306
PLOTSPEC, 325
Plotting a moment tensor solution, 351
Plotting continuous data, 162
Plotting epicenters, 209, 211
Plotting fault plane solutions, 351
Plotting pictures, 105
Plotting place names, 204
Plotting response files, 501
Plotting traces, 99
Pn phase, 111
png file, 105
Polarity, 156, 165, 336
Polarity change, 269
Polarity reading, automatic, 338
Polarity, check, 338
Polarity, correct, 269
Polarity, relative weight, 338
Polarity, select for, 231
Poles and zeros, 294, 304, 554
Polygon, 469
Polygon, select, 232
Polygon, select in, 204
Postal code, 518
Postscript, 21, 146
Power spectral density, 446
Power spectrum, 178, 181
PR_RESP, 501
Preprocessing of data, 164, 194
Presp, 201
Presp.eps, 506
Print S-file, 99
Print.out file, 131
Printer, 21
Printing, 104
Printing on PC, 24
Probabilistic seismic hazard analysis, 451
Probable explosion, 230
Problem HYP, location, 133
Problem HYP, Moho not found, 127
Problem MULPLT extract, 263
Problem UPD, 244
Problem WAVETOOL, zero output, 263
Problem, backup files in EEV, 94
Problem, Bouchon, 425
Problem, EPIMAP, 206
Problem, file name length, 17
Problem, filelist, 216
Problem, HYP, 118, 121, 124
Problem, MULPLT area, 204
Problem, MULPLT spectra, 182
Problem, no location, 117
Problem, PC, 25
Problem, picking amplitude, 164
Problem, q-correction, 180
Problem, RESP, 502
Problem, rotation, 182
Problem, Rotation and removing response, 170
Problem, select, 234
Problem, UPD, 244
Problem, UPDATE, 244
Problem, VELEST, 476
Problem: Do not find waveform file, 13
<table>
<thead>
<tr>
<th>Problem</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response file naming</td>
<td>508</td>
</tr>
<tr>
<td>S-file changed name</td>
<td>22</td>
</tr>
<tr>
<td>Problems in SEISAN</td>
<td>106</td>
</tr>
<tr>
<td>Profile, hypocenters</td>
<td>204</td>
</tr>
<tr>
<td>Profile_num</td>
<td>206</td>
</tr>
<tr>
<td>PSD</td>
<td>257</td>
</tr>
<tr>
<td>PSENI</td>
<td>294</td>
</tr>
<tr>
<td>Public Seismic Network</td>
<td>294</td>
</tr>
<tr>
<td>Python</td>
<td>25</td>
</tr>
</tbody>
</table>

| Q as a function of depth | 176 |
| Q correction in SEISAN | 175 |
| Q, determine by spectral fitting | 181 |
| QNX | 294 |
| QNXSEI | 294 |
| Qstat | 404 |
| QuakeML | 282, 299 |
| QuakeMl | 282 |
| Quality factor | 383, 400 |
| Quality factor, correct for | 443 |
| Quality factor, determine by spectra | 443 |
| Quality factor, determine with coda | 383 |
| Quanterra | 293 |

| Rake | 341 |
| Raster file | 220 |
| RDSEED | 296 |
| Rdseed | 266 |
| rdseed.SEISAN.DEF | 77 |
| rdseed.STATION0.HYP | 77 |
| rdseed2seisan | 77 |
| REA directory | 11 |
| REA.LOG | 85 |
| Read S-file | 528 |
| Recover files from the DELET database | 93 |
| Reference station | 97 |
| REFTFTEK | 287, 298 |
| REG_KEEP_AUTO | 163 |
| Register event | 256 |
| Register event, keep phases | 37 |
| register run process | 35 |
| Registration | 145 |
| Registration and preprocessing | 164 |
| Relative spectra | 445 |
| Relative weight of polarity | 338 |
| Rename event type | 99 |
| Reorder Hypocenters | 274 |
| REPORT | 180, 249 |
| Report.inp | 256 |

| Report.out | 250 |
| Report_n.out | 250 |
| Reporting agency | 127 |
| RESAMP | 266 |
| Resampling | 266 |
| RESET TEST | 127 |
| Residual weight | 129 |
| Resolution | 106 |
| RESP | 77, 499, 501 |
| RESP response file format | 563 |
| Response file | 17, 199, 554 |
| Response in header | 554 |
| Response removal | 445 |
| Response, GSE | 293 |
| Rg phase velocity | 128 |
| RMS as a function of depth | 100 |
| RMSDEP | 100 |
| Rotated seismograms | 169 |
| RSAM | 263 |
| RSASEI | 295 |
| RT_SEIS | 287 |
| RTU.INI | 295 |

| S phase | 111 |
| S-file | 11, 17, 67, 70, 239 |
| S-file changed name | 22 |
| S-file, error | 94 |
| S-file, parameters | 528 |
| S-file, print | 99 |
| S-file, update | 100 |
| S-files, collecting | 239 |
| S-P and L-S differences | 117 |
| SAC | 12, 303 |
| SAC response | 501 |
| SAC response file format | 563 |
| SACLIB | 21 |
| Sample rate | 267 |
| SAMPLE_Graphics | 529 |
| sample_instrument_correction | 529 |
| SAMPLE_READ_ARC | 528 |
| SAMPLE_READ_CONT | 528 |
| SAMPLE_READ_WAV | 528 |
| SAMPLE_UDP_MESSAGE | 529 |
| Sample_write_wav | 528 |
| SE local data base, waveform files | 51 |
| SE working directory | 51 |
| Search | 100 |
| Search by day and hour | 93 |
| Searching for text string in nordic files | 234 |
| Searching headers in waveform files | 268 |
INDEX

Searching in the database, 227
SEED, 12 296
SEED data, time gap, 146
Seed reading, 143
SEED response, 501
SEED, extract data, 266
SEI2PSXY, 295
SEIASC, 267
SEIASC format, 551
SEIBINASC, 529
SEICUT, 267
SEIDEL, 268
Seidim.inc, 22
SEIGMT, 283
SEIM88A, 296
SEIPITSA, 296
SEIS2VIEWER, 224
SEISAF, 296
SEISAN ASCII format, 551
SEISAN continuous data, 16
Seisan Explorer User Guide, 41
Seisan extension, 21
SEISAN waveform file, 17
SEISAN waveform files to or from ascii, 267
SEISAN.bash, 21
SEISAN.csh, 21
SEISAN.DEF, 31
SEISAN.DEF in SE, 12
SEISAN2MSEED, 296
SEISAN_EDITOR, 21
SEISAN_TOP, 21
SeisanExplorer source code, 29
SEISANPR, 24
SEISARCH, 21
SeiscomP archive data, 14
SEISEI, 268
SeisGram, 296
SEISLOG, 294
Seismic design spectra, 151
Seismic hazard analysis, 452
Seismic moment, 72
Seismic moment, average, 132
Seismic noise, 181
Seismic risk related programs, 151
Seismic source zones, 451
SeisimityViewer, 283
Seismograph, 149
Seismometer constants, 403
SEISNET, 70 163
SELECT, 12 22 103 227
Select epicenters in an area, 206
Select events from database, 93
Select for phase, 231
select.inp, 233
Select.out, 227
SELECTC, 234
Selection in polygon, 204
Selection on errors, 231
SELMAP, 225 283
SELSEI, 268
Serial number, instrument, 284
SESAME, 296
SFORMAT, 528
Sg phase, 111
SGRSEI, 296
Short user guide, 68
Signal duration, 315
Signal, damped, 530
Signal.out, 171
SILSEI, 298
Simulate drum recording, 149
Single channel files, 268
Single files, 11
Single station location, 111
Single trace mode, 161
Sismalp, 298
SISSEI, 298
Site effects, 451
Size of graphics window, 27
Sn phase, 111
Soil amplification, 443
Sonic boom, 230
Sorting in MULPLT, 381
Source displacement spectrum, 175
Source parameters, 175
Source radius, 177
Source time function, 428
Spain, 282 295
SPEC, 443
spec.out, 178
Spec.par, 444
spec_amps.out, 444
Spectral analysis, 443
Spectral fitting, 180
Spectral model, 176
Spectral output file, 444
Spectral parameters, 193 250 315 143 462
Spectral parameters, search for, 232
INDEX

Spectral parameters, update, 244
Spectral ratio, 443
Spectrogram, 182
Spherical harmonic station corrections, 126
SPLIT, 78, 239, 241
Split an event, 93
Sprengnether, 292
SSA, Kinematics, 291
SSR, Kinematics, 291
STA/LTA, 315
Stand-alone programs, 105
Start and end date, 104
Start depth, 120
Start time, 26
Starting location, 120, 129
STASEI, 283
Station and channel codes, 505
Station azimuth, 174
Station coordinates, 26, 277, 283
Station listing in print.out, 134
Station selection, 277
Station, only 4 characters, 276
STATION0.HYP, 26, 129
STATION0.HYP in SE, 42
STATIS, 251
Statistics, 251, 252
STATSTAT, 253
Storing magnitudes, 72
Stress drop, 177, 180, 235
Strike, 341
Subroutines in SEISAN, 531
Subset of database, 104
SUD2ASC, 298
SUDS, 298
SUDS format, 556
SUDSEI, 298
Sun and PC differences, 13, 596
Surface wave, 115
Surface wave analysis, 178
Surface wave dispersion, 176
Swarm, identify, 253
Swarm, seismic, 253
Swarm.out, 253
Synchronize S-file name and ID with origin time, 100
SYNT, 125
synt_seis.out, 381
Synthetic seismograms, 423
Synthetic test of FOCMEC, 341
T and P-axis, 338
T-phase, 128
Terra Technology, 298
TERSEI, 298
Test graphics, 529
Test mouse, 529
Test programs, 527
Test signals, make, 529
Text string search, 232
Theoretical amplitudes for FOCMEC, 341
Theoretical phases, 95
Three component plotting, 197
Threshold magnitude, 453
Time correction, 269
Time delay, 269
Time gap, 146
Time gaps, 143
Time shift data, 269
Time, uncertain flag in waveform file, 294
Timing error, 149, 156
Timing indicator, 554
TMP, 13
Top directory, 21
Trace plotting, 49, 143, 197, 295
TRACE_PLOT, 197
Travel time error, 129
Travel time outlier, eliminate, 119
Trigger, 328
Tsd.out, 269
TSIG, 529
TTLAYER, 434
TTLOCAL, 434
TTTTLT, 100, 334
tttplot.eps, 434
tttplot.out, 434
UDP, 529
Uncertain time, 156
Uncertain timing flag, 284
Unformatted read and write, 555
Units, 176, 502
Unknown type, 230
UPD, 244
UPDATE, 86, 109, 102, 143, 180, 243, 244
Update header line, 117
Update spectral parameters, 244
Update without relocation, 244
Updated S-files, 109, 243
Upper and lower case, 15
UPPER case, 165
INDEX

USERCOM, 100
USGS, 82, 283
USGSNOR, 283
Using azimuth phases, 128
Using c, 528

VELEST, 298, 473
Veltest for Windows, 173
VELMENU, 473
VELNOR, 298
Velout.dif, 475
Version, 31
Vincenty, 530
Volcanic, 230
Volcanic event, 163
Volcanic tremor, 489
VOLCANO.DEF, 489
VOLCSTAT, 489, 490
Vp/Vs ratio, 127
Vp/Vs, calculate, 439

W_EMAP, 211
WADATI, 149
Wadati, 101
WAV data base, 13
WAV directory, 12
WAVASC, 298
Wavefor to ASCII file, 298
Waveform conversion programs, 284
Waveform file format, 551
Waveform file name, 269
Waveform file structure, 559
Waveform file, cut, 267
Waveform file, delete, 170
Waveform file, extract from, 171
Waveform file, fix headers, 269
Waveform file, split, 268
Waveform files, 12
Waveform files, join, 256
Waveform files, merge, 170
Waveform.for, 528
Waveform.inc, 528
Waveform_names.out, 233
WAVETOOL, 260
WAVFIX, 255
Wavfix.tim, 269
WAVFULLNAME, 272
WEB options, 234
webinar, 1, 68
Weigh key, 165
Weight, 118, 128, 165
WGS84, 530
WGSSEL, 298
Wildcard, 245
WKBJ, 423, 424
Wood Anderson, 461
Working directory in SE, 91
WORLD.MAP, 26
Write S-file, 528
Wrong S-file name, 22

X Windows, 21
XCLUST, 521
Xnear and xfar, 127

Y5DUMP, 294
Yearly number of events, 252

Zoom in MULPLT, 155